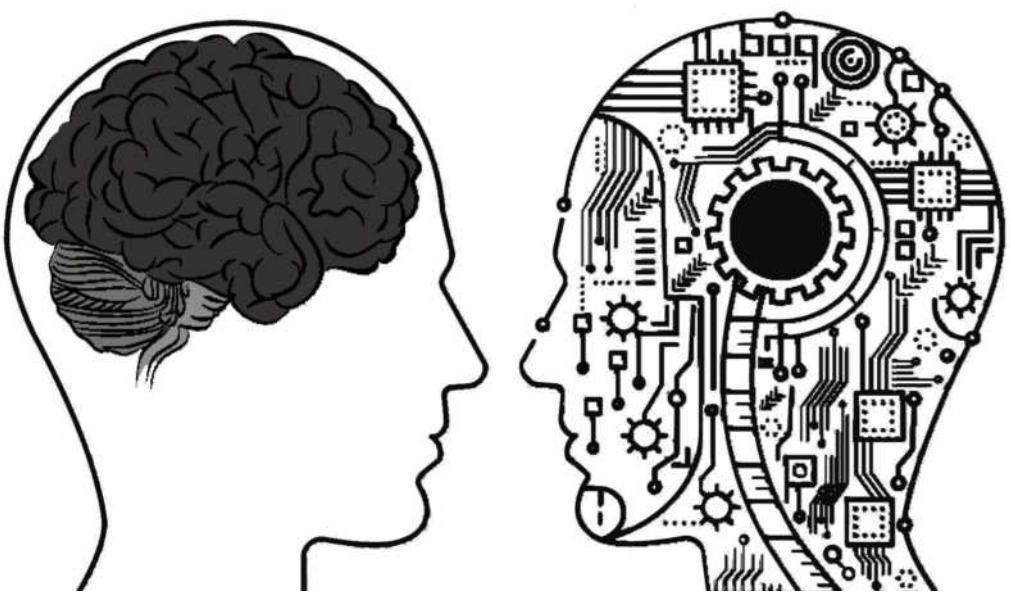
Learning Probabilistic interaction models

Multisensorial data driving situational and self-awareness models in autonomous agents



Mohamad Baydoun

UNIVERSITÀ DEGLI STUDI
DI GENOVA

Scuola Politecnica - Ingegneria
Dipartimento di Ingegneria Navale, Elettrica, Elettronica e delle Telecomunicazioni - DITEN

UNIVERSITÀ DEGLI STUDI
DI GENOVA

Learning Probabilistic interaction models

Multisensorial data driving situational and self-awareness models in autonomous agents

by

Mohamad Baydoun

A thesis submitted for the degree of
Doctor of Philosophy

Scuola Politecnica - Ingegneria
Dipartimento di Ingegneria Navale, Elettrica, Elettronica e delle Telecomunicazioni - DITEN

February 2020

"Do for this life as if you live forever,
do for the afterlife as if you die tomorrow."

- Ali ibn Abi Talib

Gratitude

First of all, I would like to thank my family. It is because of their never ending support that I have had the chance to progress in life. Their dedication to my education provided the foundation for my studies.

I would like to especially thank Professor Carlo Regazzoni and Professor Lucio Marcenaro because of their authentic way of leading, patience and dedication through these years. Without their leadership and guidance, this work would not have been possible. It has been an honor to have the opportunity to discuss and share ideas during this process. I am also deeply grateful to Professor Andrea Cavallaro from the Queen Mary University of London for his supervision and availability for my first two years. His willingness to give his time so generously has been very much appreciated.

I want to thank the co-authors of published papers, especially to Damian Campo, Mahdyar Ravanbakhsh and the research group from the University Carlos III in Madrid Professor David Martín and Pablo Marín for their collaboration.

I would like to acknowledge my colleagues, with whom I have shared moments of deep anxiety but also of big excitement. Their presence was very important in for the stimulating discussions, for the sleepless nights we were working together before deadlines. In particular, Andrea Toma, Vahid Bastani, Oscar Urizar, Ali Krayani, Muhammad Farrukh, Hafsa Iqbal, Hassan Zaal and Divya Kanapram.

I can't forget my friend Ali Abou Khalil, with whom I shared difficult times. He encouraged me and celebrated every result that I achieved.

Last but not least, I would especially like to thank my wife and daughter. They have been extremely supportive for me throughout this entire process and has made countless sacrifices to help me in getting to this point.

Abstract

We live in a multi-modal world; therefore it comes as no surprise that the human brain is tailored for the integration of multi-sensory input. Inspired by the human brain, the multi-sensory data is used in Artificial Intelligence (AI) for teaching different concepts to computers.

Autonomous Agents (AAs) are AI systems that sense and act autonomously in complex dynamic environments. Such agents can build up Self-Awareness (SA) by describing their experiences through multi-sensorial information with appropriate models and correlating them incrementally with the currently perceived situation to continuously expand their knowledge. This thesis proposes methods to learn such awareness models for AAs. These models include SA and situational awareness models in order to perceive and understand itself (self variables) and its surrounding environment (external variables) at the same time. An agent is considered self-aware when it can dynamically observe and understand itself and its surrounding through different proprioceptive and exteroceptive sensors which facilitate learning and maintaining a contextual representation by processing the observed multi-sensorial data.

We proposed a probabilistic framework for generative and descriptive dynamic models that can lead to a computationally efficient SA system. In general, generative models facilitate the prediction of future states while descriptive models enable to select the representation that best fits the current observation. The proposed framework employs a Probabilistic Graphical Models (PGMs) such as Dynamic Bayesian Networks (DBNs) that represent a set of variables and their conditional dependencies. Once we obtain this probabilistic representation, the latter allows the agent to model interactions between itself, as observed through proprioceptive sensors, and the environment, as observed through exteroceptive sensors.

In order to develop an awareness system, not only an agent needs to recognize the normal states and perform predictions accordingly, but also it is necessary to detect the abnormal states with respect to its previously learned knowledge. Therefore, there is a need to measure anomalies or irregularities in an observed situation. In this case, the agent should be aware that an abnormality (i.e., a non-stationary condition) never experienced before, is currently present.

Due to our specific way of representation, which makes it possible to model multi-sensorial data into a uniform interaction model, the proposed work not only improves predictions of future events but also can be potentially used to effectuate a transfer learning process where information related to the learned model can be moved and interpreted by another body.

The contents of this thesis are based on several peer-reviewed conference and journal papers published during my PhD studies together with some works presented in workshops and partial results of projects that are under preparation for publishing in the coming period. This thesis is divided into five chapters:

Chapter 1 - Introduction: This chapter states the general topic of the thesis and also provides a review of the literature related to the topic and justifies the research contribution presented in this document. It also presents the list of published papers during my PhD.

Chapter 2 - Single-Modality State Representation and Abnormality Detection: This chapter presents a computational approach that facilitates the representation and modeling of agents' dynamic behaviors and the detection of new experiences using only positional information, as a simple awareness system. This chapter shows how Gaussian Processes (GP) and Super-Pixel (SP) techniques can be employed for learning models later used by a bank of Kalman filters for inference purposes. Finally, this chapter supports with experimental results for detecting anomalies and trajectory classification.

Chapter 3 - Multi-Sensorial Data for Learning a Multi-Modal Awareness System: This chapter introduces two approaches to learn multi-modal SA models by using different levels of supervision for extending the prediction and abnormality detection into a multi-level fashion (discrete-continuous information). Namely, we first propose a semi-supervised GP-based approach, and then an unsupervised incremental learning process is introduced. This chapter describes how the clustering techniques such as Self-organizing Maps can be employed for learning a multi-level DBN that describes observed data. Additionally, in this chapter we introduce a new representation of Markov Jump Particle Filter (MJPF) that facilitates the prediction and detection of abnormality in the proposed DBN.

Chapter 4 - A Unified Interaction Multi-Modal Awareness System: This chapter presents a method for modeling causality between multi-sensorial information as an interaction level between different modalities or entities. In particular, we explain how it is possible to integrate several DBNs into a unified coupled probabilistic model that can be used to make inferences of multi-sensory data. Accordingly, it is shown how the MJPF can be adequately employed for multi-sensorial scenarios by considering the possible dependencies between different data sources. In addition, this chapter compares two different approaches for modeling the interaction of multi-sensorial information.

Chapter 5 - Conclusions and Future Work : This chapter concludes the thesis and lists the advantages. Additionally, possible future directions of the research in question are here stated and discussed.

Contents

List of Figures

List of Tables

Abbreviations

AI	**A**rtificial **I**ntelligence
DBN	**D**ynamic **B**ayesian **N**etwork
FPV	**F**irst **P**erson **V**iew
GAN	**G**enerative **A**dversarial **N**etwork
MJPF	**M**arkov **J**ump **P**article **F**ilter
PGM	**P**robabilistic **G**raphical **M**odel
PL	**P**rivate **L**ayer
SP	**S**uper-**P**ixel
OS	**O**ver-**S**egmentation
UMKF	**U**n**M**otivated **K**alman **F**ilter
SOM	**S**elf **O**rganizing **M**ap
C-DBN	**C**oupled **D**ynamic **B**ayesian **N**etwork
GS	**G**eneralized **S**state
GNG	**G**rowing **N**eural **G**as
AA	**A**utonomous **A**gent
EC	**E**nvironment **C**entered
GP	**G**aussian **P**rocess
KF	**K**alman **F**ilter
ML	**M**achine **L**earning
CL	**C**ontrol **L**ayer
SA	**S**elf-**A**wareness
SL	**S**hared **L**ayer
MKF	**M**otivated **K**alman **F**ilter
PF	**P**article **F**ilter
HMM	**H**idden **M**arkov **M**odel

Chapter 1

Introduction

AAs can be seen as computational systems that interact independently, sense and act autonomously with its environment via its own sensors, and by doing so realize a set of goals or tasks for which they are designed [1–3]. This generally acknowledged relationship between an agent and its environment is schematically depicted in Figure 1.1. Generally speaking, intelligent agents continuously perform different functions: perception of dynamic conditions in the environment, action to affect conditions in the environment, reasoning to interpret perceptions, solve problems, draw inferences, and determine actions [4–6]. Thus, in this sense humans and most animals can also be regarded as AAs. One of the ultimate goal for AI systems is to construct AAs capable of human-level performance [7, 8]. However, a look at the state of current research reveals that we are quite far from achieving this goal but progressing.

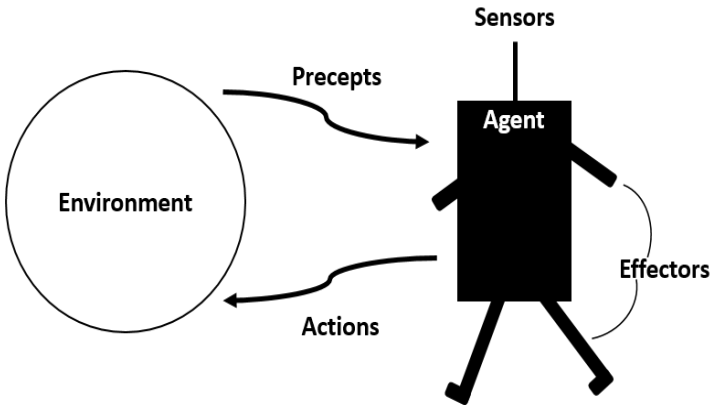


FIGURE 1.1: An Agent interacting with its surrounding environment through sensors and effectors.

AAs are developing too fast, and there is a need to improve their perception and their understanding about the environments and themselves in order to have more reliable agents. This could lead to have the agents which are smarter to perform tasks better and more importantly adapt themselves with the dynamic changing real-world environments. An agent can perceive its external world and itself by using a set of sensors. Accordingly, this set of multi-sensorial information can be divided in two main parts: exteroceptive and proprioceptive. Proprioceptive sensors measure the internal agent's parameters whereas exteroceptive sensors observe the agent's environment (see Figure 1.2).

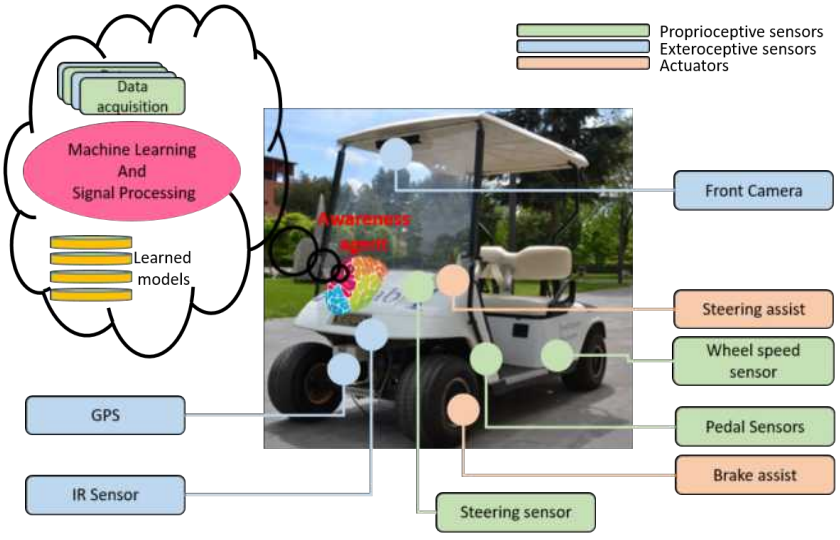


FIGURE 1.2: Physical architecture for an awareness AA. The autonomous vehicle observes the surrounding environment with exteroceptive sensors (blue) and its internal state with proprioceptive sensors (green) and translates its autonomous decisions into actions through the actuators (in red). The SA core is able to forecast the next state of the environment and of the system itself to detect anomalies and execute the derived actions.

The understanding and making inference from such information is essential to fully describe the awareness of a system. Accordingly, based on sensory data, two main awareness models could be considered:

- *Situational awareness*: which refers to model, perceive and understand the environment from the agent.
- *Self-awareness*: which allows an agent to model, perceive and understand itself (internal parameters 'effectors-related measurements of the agent').

For generating artificial aware agents, it is essential to embed the sense of self-awareness (understanding of own states) and situation awareness (comprehension of external surrounding states) in the agent in question. Figure 1.3 proposes a diagram that explains the requirements for a full awareness system.

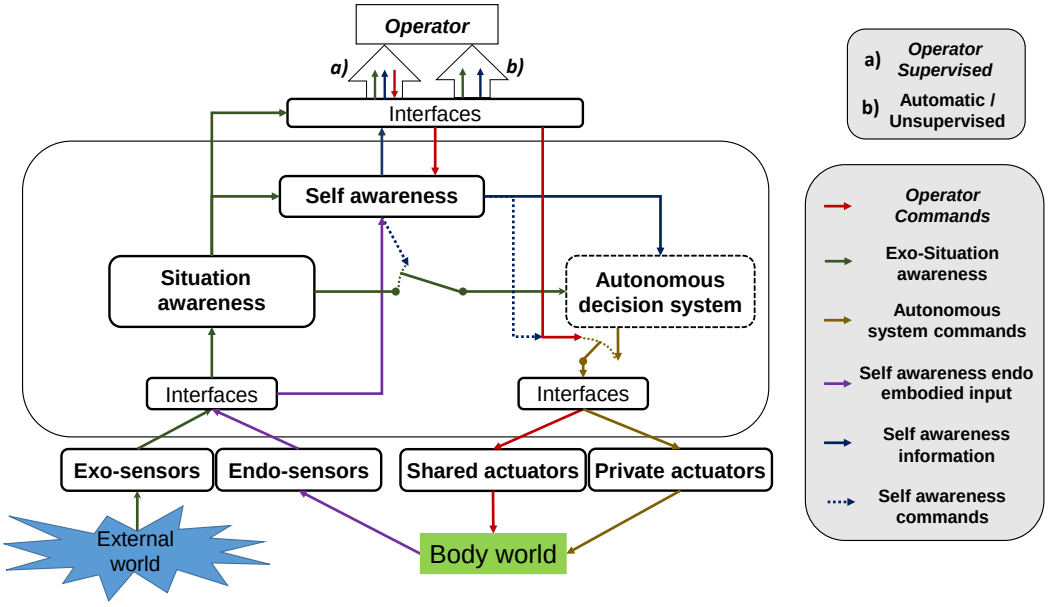


FIGURE 1.3: Proposed fully awareness diagram.

By refering to Figure 1.3, the observable *world* can be divided into two main parts:

- *External world*: It refers to the agent's surroundings that it is capable of perceiving through its exo-sensor, i.e., sensory devices dedicated to measuring the outside world or the environment where the agent is immersed.
- *Body world*: It refers to the agent's physical actuators. Accordingly, it is important to take into account the sensors dedicated to measuring the control parameters of the agent, i.e., endo-sensors and the orders given to the actuators.

Interfaces that facilitate the estimation of current environment state (exo-situation awareness) and internal states (SA) of the agent in question are required. Those interfaces enable to transform sensory data to state variables that describe the internal and external situations where a given agent is immersed. Additionally, an autonomous decision system is considered in charge of controlling the agent's actuators according to known previous learned models that guarantee a stable situation with respect to a particular goal to be accomplished. Such autonomous

system must take into consideration both exo-situation and SA states to make decisions about the next agent's action. When an anomaly in any of both awareness modules is detected, an intervention from an operator who takes control of the current situation such as described in the case of ***a)*** in Figure 1.3 takes place. In the case of ***b)***, the operator can observe the current situation but cannot take control of the agent's actions. However, in both cases (***a***) or (***b***), an interface is necessary that transforms information of exo-situation and SA into a language that can be readable and potentially manipulated by an operator.

Exo-awareness information goes to the SA module and it is the latter that provides commands to the entire system when there is an abnormality that the current models cannot handle. In such cases, the autonomous system should enter into an exploration phase to deal with the new situations or give control to the operator and learn from him the correct actions to be done in such unusual conditions.

In this thesis, I present and formulate diverse approaches to model an awareness system for AAs which are able to represent the agent's dynamics by taking into consideration the interactions between a moving agent and its surrounding environment. In particular, in this set of work, I consider a probabilistic framework in terms of Bayesian representation in order to model agent's dynamic behaviours from different sensorial information. Accordingly, signal processing and Machine Learning (ML) techniques are utilized to design an awareness model that facilitates the similarities between current realizations and previous experiences related to a given executed task. The capability of predicting the task's evolution in normal conditions (i.e., when the task follows the rules learned in previous experiences) and jointly detecting abnormal situations allows autonomous systems to increase their awareness and the effectiveness of the decision making. The proposed representation describes the system dynamically in a holistic way by considering different levels of abstractions.

1.1 Background

SA is a broad concept which describes a cognitive property of a biological—typical human—agent. At a rather abstract level, SA can be defined as the capacity to become the object of one's own attention which arises when an agent focuses not

only on the external environment but also on the internal milieu. The agent becomes a reflective observer and processing self-information. It becomes aware that it is awake and actually experiencing specific mental events, emitting behaviors, and possessing unique characteristics [9]. Another classic SA definition is proposed by Fenigstein et al. [10] who state that a self-aware agent may focus on private or public self-aspects. Private self-aspects relate to externally unobserved events and characteristics such as emotions, physiological sensations, perceptions, values, goals and motives, whereas public self-aspects are visible attributes such as behavior and physical appearance.

Over the years, SA has been an object of intensive discussions and studies in different disciplines such as philosophy, psychology and cognitive sciences (e.g., [11–13]). Common aspects of the proposed approaches lie on the conception of SA as *i)* a cognitive embodied process composed of representational and inferential processes of an agent situated in an environment, and *ii)* an agent's property which emerges in various forms including the extent of the SA capabilities ("levels") [9, 14] and the scope of the processed information ("private and public") [10, 15]. More recently, SA concepts have been transferred to artificial systems aiming at either designing intelligent agents or analyzing their behavior. The driving motivation for the transfer of biological SA concepts to artificial systems is to improve autonomy, robustness and scalability and has been investigated in different fields including software engineering, machine learning, and robotics [16–19]. A fundamental challenge in most of these approaches is how to systematically integrate SA capabilities into artificial agents.

Moreover, SA has already been proposed for autonomic computing as a means to cope with complexity [20]. SA refers to a system's capability to recognize its own state, possible actions and the result of these actions on the system itself and on its environment. This principle has been investigated on different system layers, for example the on recently presented in the context of the Internet of Things [21–23]. However, in order to meet the complex requirements of AA, SA must not be taken separately on each layer (i.e. sensors, communication, effectors, etc.) but combined into a coherent agent SA which prevents destructive behavior due to conflicting decisions [24].

The analysis of situations and surroundings based on single modality observed dynamics is an important area in which research is advancing [25–30]. By recognizing and characterizing the context according to the movement of agents, it is possible

to improve the SA of environments [31]. By doing so, the capability of predicting future actions and conditions of the external agents is improved in return. The latter facilitates the automatic estimation of possible actions according to a given context which in turn constitutes a fundamental component for smarter systems that could predict complex scenarios given spatial trajectory information [32–36]. As pointed out in [27], contextual/semantic interpretation of observed trajectories includes information about the agent's surroundings and its own situation.

Most of the time, the single modality awareness systems lack the robustness and reliability required in several real-word applications [37, 38]. In fact, the world comprises a large amount of information which is cataloged in different sensor modalities. Processing multi-sensory information is part of our daily routines and has proven to have a direct impact in our behavioral outcomes [39, 40]. As described in [41], a multi-sensory brain allows us to combine and integrate multi-modal information, facilitate the development of cognitive skills such as extracting speech information from visual cues [42, 43] and integrate gustatory and olfactory cues for perceiving flavor [44, 45]. The examination of interactions among different sensory modalities has been a key aspect for understanding how multi-sensory brains learn from experiences and react to new ones. Accordingly, as discussed in [46], research works involving both human and nonhuman subjects have been conducted to understand how multi-sensory interactions enable behavioral, perceptual, and cognitive abilities.

Motivated by the discovered advantages brought by multisensorial processing in humans and other living beings [47–49], researchers have developed the theory of multi-sensory learning which supports the idea that brains learn and operate optimally in multi-sensory scenarios. Such an assumption is quite rational since we are constantly surrounded by multimodal stimuli that affect our behavior continuously. As discussed in [50], a given task is mastered with less effort when multi-sensory cues are available, suggesting a brain's natural preference when learning and operating with multi-sensory information. Consistently, multi-sensory brains should follow a multi-sensory protocol to elaborate perceived cues. Such a protocol considers the different sensory modalities of the brain not as independent processes but rather as a multi-sensory interactions in a contextual environment [51]. For doing so, it is necessary to consolidate the information from simultaneously experienced unisensory modalities into a single interaction multi-sensory perception [52, 53].

In light of the above, and motivated by [54], in this thesis, we consider the SA modeling from a sensor data and signal processing perspective. We propose an interaction cross-modalities structure (i.e., internal and external sensory data). That takes place in the contexts of real-life problems where information is combined from various modalities (e.g. vision and language [55, 56]) or different domains (e.g. brain and environment [57]). Having the perception from the external observer available, the internal body information conveyed by these external observations would be complementary. Moreover, given different perspectives we can model the causality between several modalities. This can be done through an interaction representation of different perspectives. We represent a SA model obtained by jointly and dynamically analyzing the sensory data endows the agent with introspection at different hierarchical levels. Such representation allows the agent to model:

- Single modality awareness system.
- Multi-modal awareness system.
- Interactions between itself as observed through proprioceptive sensors and the environment as observed through exteroceptive sensors.

Furthermore, learning new concepts dynamically is a crucial ability for an AA, and by far is the most studied type of learning in AI [58]. The incremental learning problem is the matter of learning new concepts knowledge or tasks while not forgetting old knowledge [59–64]. In this thesis, we present an approach based on the incremental learning of new dynamic models from data acquired along with agent experiences. This facilities constructing more reliable SA models.

1.2 Main Contribution

This thesis is focused on designing methods to learn awareness models for an AA form multi-sensorial information by applying probabilistic techniques such as DBN.

In addition to the main target of this thesis, the novelties that it came up with are listed as follows:

i. It proposes a hierarchical bayesian representation to model the situational awareness by analysing positional information. Such representation enables the modeling of observed motions dynamically by taking into consideration causalities between moving agent and its surrounding environments.

ii. A probabilistic switching DBNs is presented to learn a multi-modal awareness incremental model from different sources (exteroceptive and proprioceptive). Such network provides a complementary information between the SA Layers. Accordingly, a hierarchical model based on MJPF is proposed to model low dimensional data. Additionally, a hierarchical model is introduced by means of a cross-modal Generative Adversarial Networks (GANs) processing high dimensional visual data. Different levels of the GANs are detected in a "self-supervised" manner using GANs discriminators decision boundaries.

iii. Two different coupled DBN architectures are proposed to model causalities between exteroceptive and proprioceptive data. Such causalities can be considered as interaction models that encode the relationship between multimodal information.

1.3 PhD publications

The following list of publications represent the outcomes of the research done over the years of the PhD concerning the published conference and journal papers:

- **Prediction of Multi-target Dynamics Using Discrete Descriptors: an Interactive Approach**, **M. Baydoun**, D. Campo, D. Kanapram, L. Marcenaro, C. S. Regazzoni, IEEE International Conference on Acoustics, Speech, and Signal Processing (**ICASSP'19**), Brighton, United Kingdom (2019).

- **Learning Probabilistic Awareness Models for Detecting Abnormalities in Vehicle Motions**, D. Campo, , **M. Baydoun**, P. Marin, D. Martin, L. Marcenaro, A. Escalera, C. S. Regazzoni, IEEE Transactions on Intelligent Transportation Systems 2019 (**T-ITS'19**). PP(99):1-13.

- **Learning a Switching Bayesian Model for Jammer Detection in the Cognitive-Radio-Based Internet of Things**, M. Farrukh, A. Krayani,

M. Baydoun, L. Marcenaro, Y. Gao and C. S. Regazzoni, IEEE World Forum on Internet of Things (**WF-IoT '19**), Limerick, Ireland (2019).

- **Dynamic Bayesian Approach for decision-making in Ego-Things**, D. Kanapram, D. Campo, **M. Baydoun**, L. Marcenaro, E. L. Bodanese, C. S. Regazzoni and M. Marchese, IEEE World Forum on Internet of Things (**WF-IoT '19**), Limerick, Ireland (2019).

- **Jammer detection in M-QAM-OFDM by learning a Dynamic Bayesian Model for the Cognitive Radio**, A. Krayani, M. Farrukh, **M. Baydoun**, L. Marcenaro, Y. Gao, C. S. Regazzoni, European Signal Processing Conference (**EUSIPCO '19**), Coruña, Spain (2019).

- **Clustering optimization for abnormality detection in semi-autonomous system**, H. iqbal, D. Campo, **M. Baydoun**, L. Marcenaro, D. Martin, C. S. Regazzoni, International Workshop on Multimodal Understanding and Learning for Embodied Applications (**MULEA '19**), Nice, France (2019).

- **Abnormality detection using graph matching for multi-task dynamics of autonomous systems**, H. Zaal, **M. Baydoun**, L. Marcenaro, L. Tokarchuk, C. S. Regazzoni, IEEE International Conference on Advanced Video and Signal-based Surveillance (**AVSS '19**), Taipei, Taiwan (2019).

- **A Multi-perspective Approach to Anomaly Detection for Self-aware Embodied Agents**, **M. Baydoun**, M. Ravanbakhsh, D. Campo, P. Marin, D. Martin, L. Marcenaro, A. Cavallaro, C. Regazzoni, IEEE International Conference on Acoustics, Speech, and Signal Processing (**ICASSP'18**), Calgary, Canada (2018).

- **Learning Switching Models for Abnormality Detection for Autonomous Driving**, **M. Baydoun**, D. Campo, V. Sanguineti, L. Marcenaro, A. Cavallaro, C. S. Regazzoni, International Conference on Information Fusion (**FUSION'18**), Cambridge, UK (2018).

- **Unsupervised Trajectory Modeling Based on Discrete Descriptors for Classifying Moving Objects in Video Sequences**, D. Campo, **M. Baydoun**, L. Marcenaro, A. Cavallaro, C. S. Regazzoni, IEEE International Conference on Image Processing (**ICIP'18**), Athens, Greece (2018).

- **Hierarchy of GANs for Learning Embodied Self-Awareness Model**, M. Ravanbakhsh, **M. Baydoun**, D. Campo, P. Marin, D. Martin, L. Marcenaro, C. S. Regazzoni, IEEE International Conference on Image Processing (**ICIP'18**), Athens, Greece (2018).

- **Learning Multi-Modal Self-Awareness Models for Autonomous Vehicles from Human Driving**, M. Ravanbakhsh, **M. Baydoun**, D. Campo, P. Marin, D. Martin, L. Marcenaro, C. S. Regazzoni, International Conference on Information Fusion (**FUSION'18**), Cambridge, UK (2018).

- **Task-dependent saliency estimation from trajectories of agents in video sequences**, D. Campo, **M. Baydoun**, L. Marcenaro, C. S. Regazzoni, IEEE International Conference on Image Processing (**ICIP'17**), Beijing, China (2017).

- **Modeling and classification of trajectories based on a Gaussian process decomposition into discrete components**, D. Campo, **M. Baydoun**, L. Marcenaro, A. Cavallaro, C. S. Regazzoni, IEEE International Conferenceon Advanced Video and Signal Based Surveillance (**AVSS'17**), Lecce, Italy (2017).

- **Hand pose recognition in First Person Vision through graph spectral analysis**, **M. Baydoun**, A. Betancourt, P. Morerio, L.Marcenaro, M. Rauterberg, C. Regazzoni, IEEE International Conference on Acoustics, Speech, and Signal Processing (**ICASSP'17**), New Orleans, USA (2017).

Chapter 2

Single-Modality State Representation and Abnormality Detection

In order to develop an autonomous system, a very first step of any awareness model is to understand the dynamics and the pattern of the changes in different sensorial modalities. In other words, modeling, understanding and predicting how dynamical systems evolve in time are important tasks for improving the estimation of future events, preventing undesired situations and building smart systems capable of interacting with the environment in an optimal way given a determined context. This chapter explains in details the theory behind the techniques used in the proposed method for modeling and understanding agents' dynamics from positional information. Additionally, it explains the proposed methodology for analyzing spatial trajectory data under a Bayesian modeling framework. As mentioned previously, the proposed method in this chapter assumes that only information about agents' location is available through time.

2.1 Understanding agents' dynamics

A moving agent needs some sort of goal information that indicates the desirable states in the environment. It keeps track of the world state as well as a set of goals it is trying to achieve, and it chooses an action that will (eventually) lead

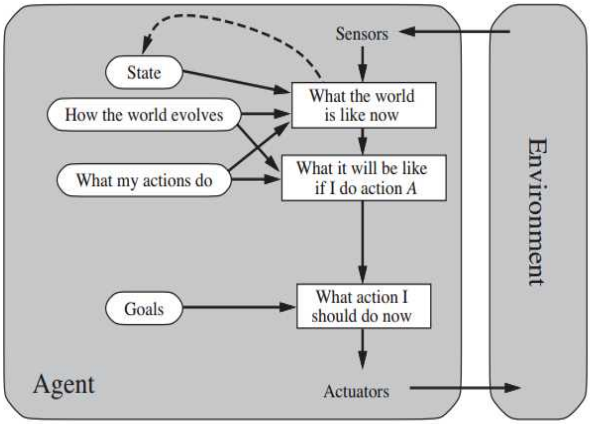


FIGURE 2.1: Schematic Diagram of the Model-based Goal-based agents, taken from [66].

to the achievement of its goals [29, 65]. Accordingly, the representation of such agents adjust to the *goal-based agents* is described in the work of Russell and Norvig [66]. Decisions made by such class of agents are based on a cognitive perception of their surroundings combined with a goal to be achieved. As shown in the diagram in Figure 2.1, when this kind of agents take a decision, their goals (principal motivation) and their surroundings (agents' perceived environment) play a fundamental role.

In this chapter, a probabilistic approach is considered to understand dynamic relationships among moving agents and their surrounding environments. Such representation paves the way to make future inferences of agents' states as proposed in in [67–70]. Accordingly, agents' motions are considered to be zones in the environment where the activity of going towards a specific goal of the scene [71, 72]. For modeling such zones, it is uses a Bayesian reasoning for interpreting and models observed data. Agents' states are organized in such a way that a GP regression can be applied to understand their dynamics depending on their location in the environment. Consequently, the next subsections focus on the technicalities associated with the Bayesian representation of positional data and the theory behind the non-parametric learning of state relationships made by the GP regression.

2.1.1 Bayesian modeling of positional information

Let $\boldsymbol{x} \in \mathbb{R}^d$ be a generalized coordinated system such that the scene is described by a d-dimensional space. The state of a given moving agent (l) is defined as a vector composed of its positions and m time derivatives, such that $X_{(l)} = [\boldsymbol{x}_{(l)} \ \dot{\boldsymbol{x}}_{(l)} \cdots \boldsymbol{x}_{(l)}^{(m)}]^T$.

This work considers temporal dependencies for each moving agent's dynamics of the type $p(X_{(l),k}|X_{(l),k-1})$, i.e., the dependence of the current states on the past information. Where $X_{(l),k} = [\boldsymbol{x}_{(l),k} \ \dot{\boldsymbol{x}}_{(l),k} \cdots \boldsymbol{x}_{(l),k}^{(m)}]^T$ represents the state of an agent (l) at a particular time instant k.

For modeling the evolution of states through time, a dynamical model that relates present and future states is proposed. Consequently, it introduces a dynamic equation that describes the agents' state transition model such that:

$$X_{(l),k} = f_A\big(X_{(l),k-1}\big) + w_k, \tag{2.1}$$

w_k represents the process noise introduced by the function $f_A(\cdot)$, A encodes the way by which an agent moves when it is affected by a certain motivation. In that sense, A indexes the identified organized motions produced by an external entity. Such variable follows the reasoning of static motivation spots described previously in [69].

Since measurements from devices are employed to infer agents' states, an observation model can be defined as:

$$Z_{(l),k} = h\big(X_{(l),k}\big) + v_k, \tag{2.2}$$

where Z_k is the agent's observation at the time instant k, v_k is the observation noise introduced by the measurement device and $h(\cdot)$ is a function that maps agents' states into observations.

Our method uses a DBN for representing and modeling situations where location measurements Z are available. DBNs are suitable for describing agents' dynamics due to their capability of modeling future instances based on observations in a probabilistic way.

2.1.2 GP regression

Done with highlighting a DBN, a GP can be defined as a statistical model where observations occur in a continuous domain, e.g., space, velocities or time. GP associates a normal distributed random variable to points in a continuous space.

GPs can be seen as a supervised ML algorithm that uses Bayesian inferences for regression or classification purposes. GPs measure the similarity between input and output data; and through a kernel function, it can predict values around observed information provided training stage. Additionally, the prediction produced by GPs contains not only estimations but also an uncertainty measurement associated with them.

A GP is fully specified by a mean and covariance functions. Such functions are defined separately, and they basically consist of a functional form and a set of hyper-parameters to be adjusted. Thus, GP can be seen as the probability distribution over the function:

$$g(\mathcal{X}) \sim GP(\mu(\mathcal{X}), \sum(\mathcal{X})), \tag{2.3}$$

where $g(\cdot)$ is distributed as a GP with mean function $\mu(\mathcal{X})$ and covariance function $\sum(\mathcal{X})$.

GPs are widely used as prior functions in nonlinear-nonparametric regressions and classification problems. The goal of GP regressions is to find a function $g(\cdot)$ that relates input $\mathcal{X}$ with output $\mathcal{Y}$ data, such that:

$$\mathcal{Y} = g(\mathcal{X}) + \epsilon, \tag{2.4}$$

where ϵ represents the estimation error, $g(\cdot)$ is a function that relates input and output data. Figure 2.2 shows a simple example of a GP regression between one-dimensional input and output variables. Blue crosses in Figure 2.2 indicate the observed information whereas the blue line represents the non-parametric function $g(\cdot)$. The gray contour captures the uncertainty of GP's estimations. Note that gray areas become wider (more uncertain) in cases where no evidence (absence of observations) is available; indicating that estimations in such points are less reliable.

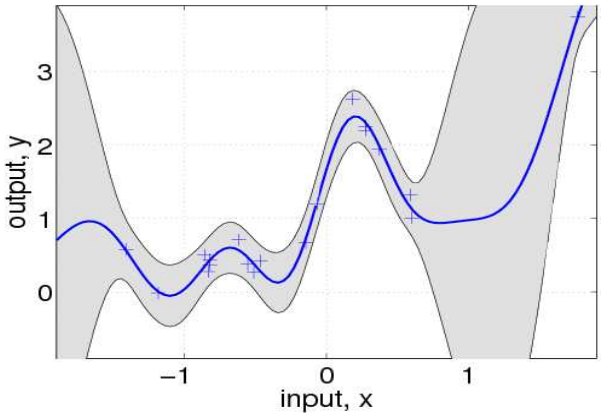


FIGURE 2.2: Example of GP regression (1-dimensional approximation).

2.2 Building of dynamical models

As it is assumed to obtain series of measured location data from agents, it is possible to propose a simple base filter that takes into account the position and dynamics of agents. Such a baseline model assumes that agents move arbitrarily around their locations due to the lack of a motivator of action [73]. From such filter's formulation, more complex filters can be obtained as observed patterns in the environment are detected. An ideal baseline filter to do such task is based on the simplest dynamical model for describing agent's actions. Accordingly, a *non-Motivated* filter is proposed as a basis for representing hierarchical motions inside a PGM structure. Table 2.1 lists the mathematical notations of the most relevant variables used in our method.

2.2.1 Non-motivated dynamical model

Let agents' states to be composed of their position and velocity such that: $X_{(l),k} = [\boldsymbol{x}_{(l),k},\ \dot{\boldsymbol{x}}_{(l),k}]^T$, where k indexes a given time instant, and (l) labels a particular moving agent. A non-motivated dynamical model (UnMotivated Kalman Filter (UMKF)) based on a random walk model is written as follows:

$$X_{(l),k+1} = FX_{(l),k} + w_{(l),k}, \tag{2.5}$$

where $w_{(l),k}$ is assumed to be drawn from a zero mean multivariate normal distribution with covariance $Q_{(l),k}$, such that $w_{(l),k} \sim \mathcal{N}(0, Q_{(l),k})$. F can be written

as:

$$F = \begin{bmatrix} I_d & 0_{d,d} \\ 0_{d,d} & 0_{d,d} \end{bmatrix}$$

where d represents the number of dimensions of the environment space $\boldsymbol{x}$, I_n represents the $n \times n$ identity matrix and, $0_{n,n}$ is a $n \times n$ square zero matrix.

As can be seen from equation 2.5, the proposed model suggests that agents will rest in a quasi-static location, and only random noise perturbations, modeled by $w_{(l),k}$, will affect their states. Such assumption implies that covariance components $Q_{(l),k}$ are small enough to model subtle random effects that an agent with no motivations can have, i.e., random oscillations around a given point.

The last assumption of the proposed non-motivated filter relies on the linear relationship between observations of agents' locations, Z, and the state of agents X. Consequently, it is assumed that:

$$Z_{(l),k} = HX_{(l),k} + v_{(l),k}, \tag{2.6}$$

where $v_{(l),k} \sim \mathcal{N}(0, R_{(l),k})$, and $R_{(l),k}$ represents the measurement covariance noise. Additionally, since Z_k is assumed to be the agent's position measurement at time

$Z_{(l),k}$	$\triangleq$	Location measurement of the agent at a time k
$X_{(l),k}$	$\triangleq$	State of agent l measurement at a time k
$U_{(l),k}$	$\triangleq$	Velocity of agent l at a time k
$\tilde{Y}^{(\mathbf{0})}_{(l),k}$	$\triangleq$	Non-motivated model's innovation for agent l at a time k
$\hat{X}^{(\mathbf{0})}_{k\|k-1}$	$\triangleq$	State prediction from a non-motivated model
$\hat{X}^{(\mathbf{1})}_{k\|k-1}$	$\triangleq$	State prediction from a motivated model
$\mathfrak{X}_A$	$\triangleq$	GP's location grid for an activity A
$\mathfrak{Y}_A$	$\triangleq$	GP's innovation (velocity) grid for an activity A
$\mathfrak{y}_A$	$\triangleq$	GP's uncertainty grid for an activity A
$\xi_{A,q}$	$\triangleq$	GP's joint uncertainty grid for an activity A
λ_{val}	$\triangleq$	Threshold to identify valid GP's information
$\mathfrak{C}^A_{\mathcal{X},n}$	$\triangleq$	Grid location information related to over-segmented region n for an activity A
$\mathfrak{C}^A_{\mathcal{Y},n}$	$\triangleq$	Grid innovation (velocity) information related to over-segmented region n for an activity A
λ_{bhat}	$\triangleq$	Threshold for merging over-segmented regions
n_k	$\triangleq$	Grown region activated at time k
A_k	$\triangleq$	Activity executed at time k

TABLE 2.1: Mathematical notations.

k, and the matrix H has the following form:

$$H = \begin{bmatrix} I_d & 0_{d,d} \end{bmatrix}$$

2.2.2 Motivated dynamical model

Effects of motivations acting on agents are modeled as a control input that influences agents' velocities (actions). In this sense, it is possible to consider the following dynamical model that encodes the motivation (goal) effects (Motivated Kalman Filter (MKF)):

$$X_{(l),k+1} = FX_{(l),k} + BU_{(l),k} + w_{(l),k}, \tag{2.7}$$

where

$$B = \begin{bmatrix} \Delta k I_d \\ I_d \end{bmatrix}$$

the parameter U_k is a velocity component that encodes the effect of surroundings. U_k can be seen as the sum of diverse motivations (goals) by which an agent is exposed such that:

$$U_{(l),k} = \sum_{m=1}^{M} u_k^{(m)} \tag{2.8}$$

where $u_k^{(m)}$ represents the motion effect produced by a motivation m. M is the total number of motivators acting on agent l. Velocity components $U_{(l),k}$ in equation 2.7 are function of the agent's position $HX_{(l),k}$ leading to:

$$U_{(l),k} \equiv U\left(HX_{(l),k}\right) \tag{2.9}$$

For agents belonging to the same class, i.e., objects with similar motion capabilities, effects acting on them are assumed to be identical for all agents so that $X_k = X_{(l),k}$.

In order to approximate the values of $U_k^{(l)}$ based on the non-motivated model, it is considered the innovation components produced by a Kalman Filter (KF) that uses such model for making inferences. Accordingly, it is possible to define those innovations as:

$$\tilde{Y}_k^{(\mathbf{0})} = Z_k - H\hat{X}_{k|k-1}^{(\mathbf{0})} \tag{2.10}$$

The parameter **(0)** indexes estimations made by a model based on the non-motivated dynamical behavior (see equation 2.5). $\hat{X}^{\mathbf{(0)}}_{k|k-1}$ stands for the agent's state prediction at a time k given the corrected state at the instant $k-1$, i.e., $\hat{X}^{\mathbf{(0)}}_{k-1|k-1}$.

In general, innovations can be seen as quantities that measure the deviation that a proposed dynamical model presents from observations. In the ideal case, $\tilde{Y}^{\mathbf{(0)}}_{k}$ tends to zero which indicates that the utilized dynamic model explains the observed agent's motions precisely. Following this reasoning, when innovations are significantly different from zero, the proposed dynamical model should be modified to describe more accurately the observed agent' motions. In such cases, effects are added as a term $BU_{(l),k}$ as indicated in equation 2.7.

Let **(1)** index the estimations made by a dynamical model based on equation 2.7. By supposing a null innovation produced by such model, i.e., $\tilde{Y}^{\mathbf{(1)}}_{k} = 0$, it is assumed that the new motivated model describes data perfectly as follows:

$$Z_k - H\hat{X}^{\mathbf{(1)}}_{k|k-1} = 0. \tag{2.11}$$

Taking into consideration that predictions made by the non-motivated model can be expressed as: $\hat{X}^{\mathbf{(0)}}_{k|k-1} \sim \hat{X}^{\mathbf{(0)}}_{(0)k-1|k-1}$; due to low Gaussian noise $w^{(l)}_k$, it is possible to write:

$$\hat{X}^{\mathbf{(1)}}_{k|k-1} \sim \hat{X}^{\mathbf{(0)}}_{k|k-1} + BU_k. \tag{2.12}$$

Furthermore, by replacing 2.12 in 2.11, it is possible to obtain an approximation of the control vector U_k through some calculations such that:

$$HBU_k \sim Z_k - H\hat{X}^{\mathbf{(0)}}_{k|k-1} = \tilde{Y}^{\mathbf{(0)}}_{k} \Rightarrow U_k \sim \frac{\tilde{Y}^{\mathbf{(0)}}_{k}}{\Delta k}. \tag{2.13}$$

From equation 2.13, it is possible to see how innovations (from non-motivated models) approximate the agents' velocities (motivated actions). By considering such term in equation 2.7, it is possible to rewrite the motivated model as:

$$X_{(l),k+1} = FX_{(l),k} + B\left(\frac{\tilde{Y}^{\mathbf{(0)}}_{(l),k}}{\Delta k}\right) + w^{(l)}_k. \tag{2.14}$$

The built dynamic model shown in 2.14 can be used for tracking agents whose current state is $X_{(l),k}$. Sparse observed positions of an agent l can be written as

$HX^{A}_{(l),k}$ (GP inputs) and their correspondent displacements as $\tilde{Y}^{(\mathbf{0}),A}_{(l),k}$ (GP outputs) where A represents an activity, i.e., moving pattern in the scene. By using such data to approximate the function $\hat{g}_A(\cdot)$, it is possible to rewrite equation 2.14 as follows:

$$X_{(l),k+1} = FX_{(l),k} + B\hat{g}_A\Big(HX_{(l),k}\Big) + w^{(l)}_k. \tag{2.15}$$

The following section explains in detail the non-parametric methodology for obtaining $\hat{g}(\cdot)$ from observed location data.

2.3 Learning of dynamical models

A strategy is proposed for describing motions of agents under a probabilistic framework by using a GP regression that facilitates the identification and characterization of zones in the environment where simple models are valid. The diagram shown in Figure 2.3 summarizes the proposed methodology for finding such zones from observed trajectory data.

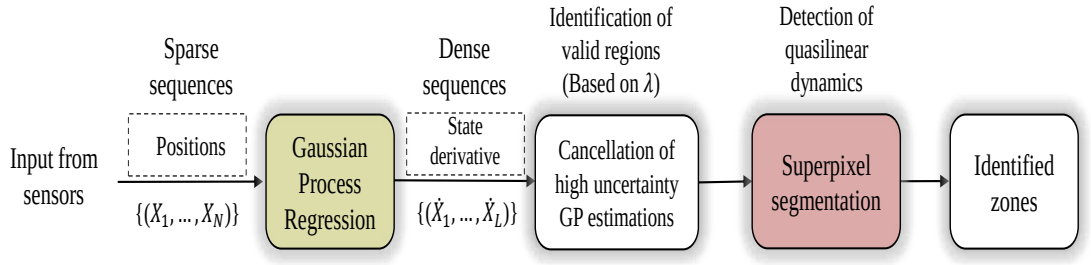


FIGURE 2.3: Block diagram of the proposed methodology for detecting zones based on GP coupled with SP.

2.3.1 GP application

Based on the local linear dynamical models calculated previously, this step aims at generalizing such models through the whole environment. Let $\mathcal{X}_A = HX^A$ be a vector consisting of a set of positions related to a given task indexed as A. Additionally, let $\mathcal{Y}_A = \tilde{Y}^{(\mathbf{0}),\boldsymbol{A}}$ be a vector of the same size of $\mathcal{X}_A$ containing the respective innovations obtained from the non-motivated model (see equation 2.10). By considering $\mathcal{X}_A$ and $\mathcal{Y}_A$ data, it is possible to use supervised learning for estimating a function $\hat{g}(\cdot)$ that relates them. Figure 2.4 shows the main idea of such a learning process.

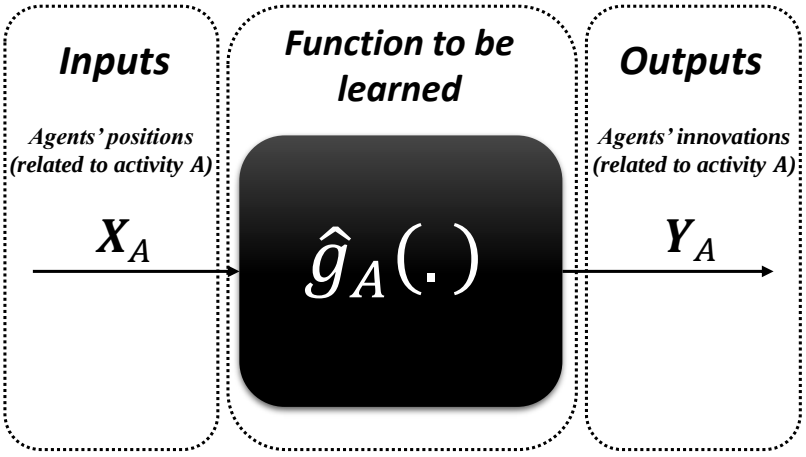


FIGURE 2.4: General scheme for supervised learning of spatial-velocity relationships for an activity A.

By taking the sparse space locations $\mathcal{X}_A$ (inputs) and their corresponding measured innovations from the non-motivated model $\mathcal{Y}_A$ (outputs), it is possible to use a GP regression that estimates the agents' motion (expected innovations) for all points in the environment when they perform a particular activity A. The following expression shows the GP regression considered in the proposed approach:

$$\mathcal{Y}_A = \hat{g}_A(\mathcal{X}_A) + \nu_A \tag{2.16}$$

$\hat{g}_A(\cdot)$ takes agent's locations as inputs and estimates their expected motions (at such positions) for an activity A. In addition, $\nu_A \sim \mathcal{N}(0, \sigma_A^2)$ is a Gaussian zero-mean white noise process. Since agents' motions are assumed to be similar at a given location when they execute a particular activity, a local Gaussian noise assumption turns out to be adequate for describing uncertainties in proposed dynamical models. Consistently, $\hat{g}(\cdot)$ is distributed as a GP defined by its mean and covariance functions (as pointed out in section 2.1.2). In this work, a linear mean and a squared exponential kernel functions are considered to perform GP estimations. Equation 2.17 shows the squared exponential kernel function as:

$$\kappa(\mathcal{X}_1, \mathcal{X}_2) = \phi^2 e^{-\frac{\|\mathcal{X}_1 - \mathcal{X}_2\|_2^2}{2\varphi^2}} \tag{2.17}$$

ϕ^2 denotes the global variance of the mapping, and φ^2 is the global smoothness parameter of the estimation. The employed kernel (covariance) allows the GP to model arbitrary nonlinear functions.

2.3.2 GP codification

Innovations estimated from the GP regression are projected on a uniformed discrete location map of the environment. In this way, GP results are discretized into three types of information:

- Spatial grid, $\mathfrak{X}_A$, which corresponds to the scene points where the GP is evaluated.
- Innovation grid, $\mathfrak{Y}_A$, which approximates the most probable motion at each evaluated position $\boldsymbol{x} \in \mathfrak{X}_A$.
- Uncertainty grid, $\mathfrak{y}_A$, which codifies the validity of GP estimations.

Accordingly, each grid data related to an activity A can be written as:

$$\begin{aligned}\mathfrak{X}_A &= \{\mathcal{X}_{A,1}, \mathcal{X}_{A,2}, \ldots, \mathcal{X}_{A,q}, \ldots, \mathcal{X}_{A,Q-1}, \mathcal{X}_{A,Q}\},\\ \mathfrak{Y}_A &= \{\mathcal{Y}_{A,1}, \mathcal{Y}_{A,2}, \ldots, \mathcal{Y}_{A,q}, \ldots, \mathcal{Y}_{A,Q-1}, \mathcal{Y}_{A,Q}\},\\ \mathfrak{y}_A &= \{\nu_{A,1}, \nu_{A,2}, \ldots, \nu_{A,q}, \ldots, \nu_{A,Q-1}, \nu_{A,Q}\}.\end{aligned}$$

Where $\mathcal{X}_{A,q}$, $\mathcal{Y}_{A,q}$ and $\nu_{A,q}$ represent input, output and uncertainty estimated information respectively associated with the grid point indexed as q (see equation 2.16). Q is the total number of cells that discretize the GP results.

The uncertainty grid $\mathfrak{y}_A$ serves to identify GP estimations that tend to be imprecise according to the training data. The GP noise of a grid point q can be represented as a Gaussian distribution $\nu_{A,q} \sim \mathcal{N}(0, \sigma^2_{A,q})$. Since an environment of d dimensions is considered; $\sigma^2_{A,q}$ is a $d \times d$ covariance matrix whose diagonal encodes the precision of GP estimations. For each grid point it is considered a joint uncertainty vector that unifies variances obtained for the d dimensions of environment such that:

$$\xi_{A,q} = \sum_{i=1}^{d} (\sigma^2_{A,q(i)}) \tag{2.18}$$

where $\sigma^2_{A,q(i)}$ represents the i-th diagonal component of the covariance matrix $\sigma^2_{A,q}$. Let $\mathfrak{s}_A$ be the set of joint uncertainties associated to an activity A such that:

$$\mathfrak{s}_A = \{\xi_{A,1}, \xi_{A,2}, \ldots, \xi_{A,q}, \ldots \xi_{A,Q-1}, \xi_{A,Q}\}$$

Additionally, let $\mathfrak{s}_A^*$ be the normalized version of the vector $\mathfrak{s}_A$ such that components of $\mathfrak{s}_A^*$ belong to the interval $[0, 1]$. Such normalization process facilitates the approximation to a beta probability distribution $\mathfrak{B}_A = beta(\alpha_A, \beta_A)$ that fits the data in $\mathfrak{s}_A^*$. Consequently, by analyzing the cumulative distribution function of $\mathfrak{B}_A$, it is possible to remove the grid points that carry high uncertainty information. A cumulative probability threshold $\lambda_{val} \in [0, 1]$ is fixed for such task. Accordingly, grid points associated with $CDF(\mathfrak{B}_A) > \lambda_{val}$ are removed in succeeding analyses. $CDF(\mathfrak{B}_A)$ represents the cumulative density function of the distribution $\mathfrak{B}_A$.

Let $\mathfrak{X}_{A,\lambda_{val}}$, $\mathfrak{Y}_{A,\lambda_{val}}$ and $\mathfrak{y}_{A,\lambda_{val}}$ be the valid GP grid data obtained by fixing a λ_{val} value. Note that as λ_{val} approaches 0, fewer valid grid points are generated. Larger λ_{val} values produce a greater number of valid GP data.

2-dimensional case is addressed here where the inputs (spatial coordinates) and outputs (spatial time derivatives) consist of two components: (x, y) and $(\dot{x}, \dot{y})$ information respectively. A GP is executed for each displacement (innovation) component and it is generalized through the spatial components of the scene. Figure 2.5 shows the GPs considered for the 2-dimensional case.

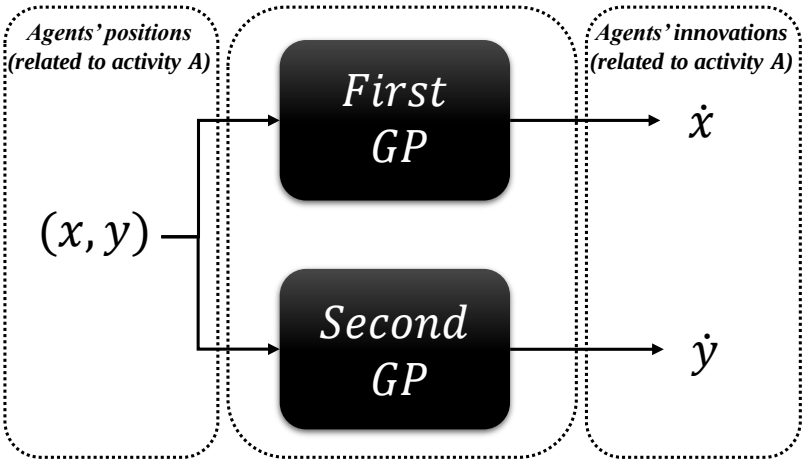


FIGURE 2.5: 2-dimensional GPs application scheme.

From Figure 2.6, it is possible to see that $\dot{x}$ components estimated through the environment are codified into a scale of Red values. Similarly, $\dot{y}$ components are codified into a range of Green values whereas the Blue channel is null. By adding the three channels together, it is possible to obtain an RGB image where pixels locations represent the spatial coordinates (x, y) of the environment and their colors encode the individuals' dynamics of a given activity. Obtained images can

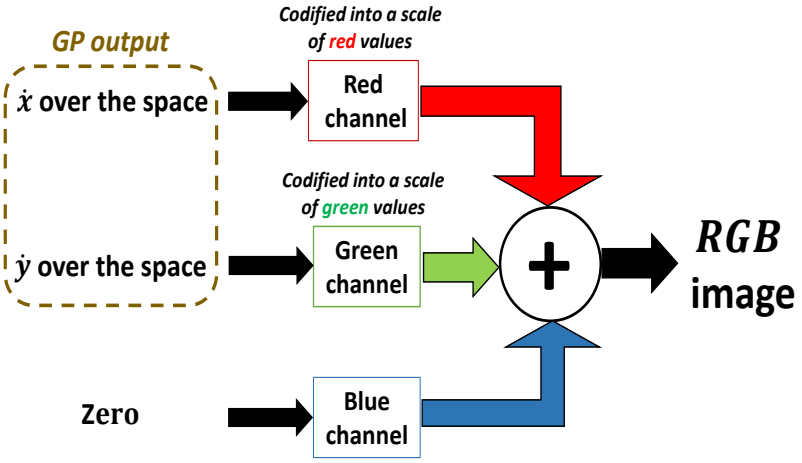


FIGURE 2.6: GPs codification into an RGB image.

be ulteriorly used for identifying models that facilitate prediction, classification, and detection of abnormalities at the time of analyzing new spatial data.

2.3.3 Identification of dynamic zones

After obtaining valid GP estimated data, i.e., $\mathfrak{X}_{A,\lambda_{val}}$, $\mathfrak{Y}_{A,\lambda_{val}}$ and $\mathfrak{y}_{A,\lambda_{val}}$, it is proposed to detect large spatial zones where agents' innovations are quasi-constant such that linear dynamic models can be applied for tracking purposes. This work adopts a SP Over-Segmentation (OS) method from which larger zones are extracted by a region growing procedure.

2.3.3.1 SP Over-Segmentation

A SP algorithm proposed by [74] is employed to discretize valid grid data into space regions where innovations are strictly similar. Accordingly, we obtain a total of N clusters (regions) that discretize vectors $\mathfrak{X}_{A,\lambda_{val}}$, $\mathfrak{Y}_{A,\lambda_{val}}$ and $\mathfrak{y}_{A,\lambda_{val}}$.

Uncertainty grid points $\mathfrak{y}_{A,\lambda_{val}}$ are not taken into consideration as an input parameter for segmenting data. Since valid data is already obtained based on such information, it is assumed that $\mathfrak{y}_{A,\lambda_{val}}$ does not influence the cluster generation.

As explained in [74], SP algorithms are based on a similarity function between two spatial points p_1 and p_2:

$$\boldsymbol{W}(p_1, p_2) = C_{\mathcal{X}}^2 \cdot \boldsymbol{W}_{\boldsymbol{\mathcal{X}}}(p_1, p_2) + C_{\mathcal{Y}}^2 \cdot \boldsymbol{W}_{\boldsymbol{\mathcal{Y}}}(p_1, p_2). \tag{2.19}$$

In this work, both points (p_1, p_2) are assumed to be part of the valid grid spatial data, i.e, $\{p_1, p_2\} \in \mathfrak{X}_{A,\lambda_{val}}$. Parameters $C_{\mathcal{X}}$ and $C_{\mathcal{Y}}$ control the relative significance of similar values in vectors $\mathfrak{X}_{A,\lambda_{val}}$ and $\mathfrak{Y}_{A,\lambda_{val}}$ respectively. $\boldsymbol{W}_{\boldsymbol{\mathcal{X}}}(\cdot,\cdot)$ and $\boldsymbol{W}_{\boldsymbol{\mathcal{Y}}}(\cdot,\cdot)$ are functions that take a couple of spatial points (p_1, p_2) and calculate the difference between their location and velocity information respectively such that:

$$\begin{aligned} \boldsymbol{W}_{\boldsymbol{\mathcal{X}}}(p_1, p_2) &= d - ||\mathcal{X}_{A,p_1}^{\lambda_{val}} - \mathcal{X}_{A,p_2}^{\lambda_{val}}||_2^2 \\ \boldsymbol{W}_{\boldsymbol{\mathcal{Y}}}(p_1, p_2) &= d - ||\mathcal{Y}_{A,p_1}^{\lambda_{val}} - \mathcal{Y}_{A,p_2}^{\lambda_{val}}||_2^2, \end{aligned} \tag{2.20}$$

where $\mathcal{X}_{A,p}^{\lambda_{val}}$ and $\mathcal{Y}_{A,p}^{\lambda_{val}}$ represent respectively the normalized position and velocity information associated to the valid point p such that $\mathcal{X}_{A,p}^{\lambda_{val}} \in \mathfrak{X}_{A,\lambda_{val}}$ and $\mathcal{Y}_{A,p}^{\lambda_{val}} \in \mathfrak{Y}_{A,\lambda_{val}}$. d is the number of dimensions of the environment.

As pointed out by [74], the vital metric to adjust is the ratio r defined as:

$$r = \frac{C_{\mathcal{X}}}{C_{\mathcal{Y}}} \tag{2.21}$$

The expected number of regions $\hat{N}$ is a key parameter to set in SP algorithm. In an OS process, the number of regions is maximized for a given ratio r. A high value of $\hat{N}$ guarantees an OS version of vectors $\mathcal{X}_{A,p}^{\lambda_{val}}$ and $\mathcal{Y}_{A,p}^{\lambda_{val}}$.

The final result of this stage consists of a set of N spatial zones where agents' innovation values are quasi-constant. Each generated region can be seen as a cluster of location and innovation data samples taken from valid information $\mathfrak{X}_{A,\lambda_{val}}$ and $\mathfrak{Y}_{A,\lambda_{val}}$. Each region is composed of two sets of data, such that:

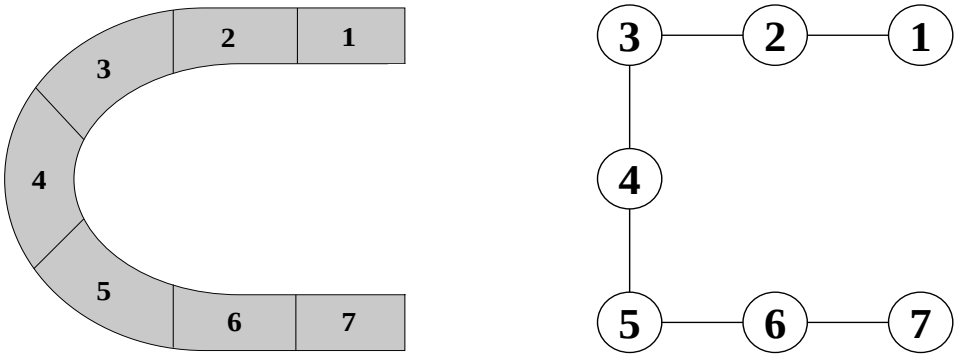
$$\begin{aligned} \mathfrak{C}_{\mathcal{X},n}^{A} &= \{\mathcal{X}_{A_n,1}^{\lambda_{val}}, \mathcal{X}_{A_n,2}^{\lambda_{val}}, \ldots, \mathcal{X}_{A_n,m_n}^{\lambda_{val}}, \ldots, \mathcal{X}_{A_n,M_n}^{\lambda_{val}}\} \\ \mathfrak{C}_{\mathcal{Y},n}^{A} &= \{\mathcal{Y}_{A_n,1}^{\lambda_{val}}, \mathcal{Y}_{A_n,2}^{\lambda_{val}}, \ldots, \mathcal{Y}_{A_n,m_n}^{\lambda_{val}}, \ldots, \mathcal{Y}_{A_n,M_n}^{\lambda_{val}}\} \end{aligned} \tag{2.22}$$

where m_n indexes the elements belonging to the region n. Additionally, $\mathcal{X}_{A_n,m_n}^{\lambda_{val}} \in \mathfrak{X}_{A,\lambda_{val}}$ and $\mathcal{Y}_{A_n,m_n}^{\lambda_{val}} \in \mathfrak{Y}_{A,\lambda_{val}}$. M_n is the total number of clustered data into the region n.

Let $\mu_{\mathcal{X},n}^{A}$ and $\mu_{\mathcal{Y},n}^{A}$ be vectors containing the average value of clustered positions and innovation components respectively. Moreover, let $\tilde{\sigma}_{A,n}^2$ be a vector containing the variances of clustered innovation components. Lastly, let $\tilde{\sigma}_{A,n(sum)}^2$ be the summation of variance components encoded in $\tilde{\sigma}_{A,n}^2$. $\tilde{\sigma}_{A,n(sum)}^2$ measures the level of linearity associated with the dynamical model in region n. Low values of $\tilde{\sigma}_{A,n(sum)}^2$

indicate coherent innovation evidence that supports the validity of a quasi-linear model in the region n.

By considering the spatial vicinity between clustered regions, it is possible to build a graph structure which encodes the location connectivity between generated regions. Accordingly, the graph's nodes represent obtained clusters whereas edges encode spatial connections between the regions. Figure 2.7 shows a straightforward example of 7 generated zones connected spatially one after the other.



(A) Superpixel generated regions. (B) Graph version of regions.

FIGURE 2.7: Example of generation of regions and graph equivalence

By applying an edge contraction operation on the superpixel's graph equivalence of obtained regions, it is possible to achieve spatial broader areas where quasi-linear motion models are still valid. As mentioned before, such a process facilitates the obtainment of extended regions containing consistent innovation information. A region growing procedure is employed for generating such broader zones which are used later for prediction and detection of abnormalities.

2.3.3.2 Region growing process

As mentioned previously, obtained regions can be mapped into a graph whose nodes contain information about average locations, i.e., $\mu^A_{\mathcal{X},n}$, dynamical models, i.e., $\mu^A_{\mathcal{Y},n}$; and their validity, i.e., $\tilde{\sigma}^2_{A,n}$.

Dynamical models can be described as a multivariate Gaussian distribution that is built based on mean values $\mu^A_{\mathcal{Y},n}$ and variances $\tilde{\sigma}^2_{A,n}$. A distance measurement ε_{n_1,n_2} between two adjacent regions n_1 and n_2 is considered to merge obtained zones such that:

$$\varepsilon_{n_1,n_2} = \sum_{i=1}^{d} D_B(P^i_{n_1}, P^i_{n_2}) \tag{2.23}$$

where $D_B(p,q)$ indicates the Bhattacharyya distance [75] between a couple of Gaussian distributions p and q. The Bhattacharyya distance helps in measuring the amount of overlap between two statistical populations and determining the closeness between probability distributions. $P^i_{n_1}$ and $P^i_{n_2}$ are Gaussian distributions of spatially adjacent regions (n_1 and n_2) related to the i-th dimension of the scene. As specified before, such Gaussian distributions are defined based on the mean and variance values in vectors $\mu^A_{\mathcal{Y},n}$ and $\tilde{\sigma}^2_{A,n}$ respectively. Additionally, consider n_{conn} to be a vector containing the set of regions that are spatially adjacent to the region n. Accordingly, in equation 2.23, $n_2 \in n_{1conn}$ and $n_1 \in n_{2conn}$.

By considering a threshold value $\varepsilon_{\lambda_{bhat}}$ for merging adjacent regions, it is possible to obtain larger zones where quasi-linear models are still valid. For fixing $\varepsilon_{\lambda_{bhat}}$, it is considered the Bhattacharyya distances (equation 2.23) between all adjacent regions. Such distances are normalized into the interval $[0,1]$, and a beta probability distribution $\mathfrak{B}_{bhat,A}$ is approximated based on such information. As it is well known, a beta cumulative distribution evaluated in the point $\varepsilon_{\lambda_{bhat}}$ provides the probability of obtaining values in the interval $[0,\varepsilon_{\lambda_{bhat}}]$. Let us define such probability as λ_{bhat}. Consistently, since the proposed distribution encodes distances between regions, probability values $\lambda_{bhat} \sim 0$ codify similar regions whereas $\lambda_{bhat} \sim 1$ capture large differences between zones. By fixing a threshold probability $\lambda_{bhat} \in [0,1]$, a maximum threshold distance $\varepsilon_{\lambda_{bhat}}$ that favors the most similar distances between regions, i.e., values in the interval $[0,\varepsilon_{\lambda_{bhat}}]$, is implicitly defined.

Couples of regions n_1 and n_2 that produce a distance measurement of the type $\varepsilon_{n_1,n_2} < \varepsilon_{\lambda_{bhat}}$ are incrementally merged such as indicated in Algorithm 1. The final result of this stage consists of larger regions that will be used for prediction and abnormality detection purposes.

Algorithm 1 Region merging process

Input:

[$\mu^A_{\mathcal{Y},n}$] Mean values of innovations in regions
[$\tilde{\sigma}^2_{A,n}$] Variance of innovation components in regions
[n_{conn}] Spatial connectivity between all regions
[$\varepsilon_{\lambda_{bhat}}$] Threshold value to fuse regions
[$\tilde{\sigma}^2_{A,n(sum)}$] Uncertanty of regions' models

Output:

[$\tilde{\sigma}^2_{A,n^*}$; $\mu^A_{\mathcal{X},n^*}$; $\mu^A_{\mathcal{Y},n^*}$] Merged regions' properties
procedure REGION GROWING
    Initialization: $\mu^A_{\mathcal{X},n^*} \leftarrow \mu^A_{\mathcal{X},n}$; $\mu^A_{\mathcal{Y},n^*} \leftarrow \mu^A_{\mathcal{Y},n}$
    $\tilde{\sigma}^2_{A,n^*} \leftarrow \tilde{\sigma}^2_{A,n}$; $\tilde{\sigma}^2_{A,n^*(sum)} \leftarrow \tilde{\sigma}^2_{A,n(sum)}$
loop:
 $n_i \leftarrow$ Region with the lowest uncertainty in $\tilde{\sigma}^2_{A,n^*(sum)}$
 $n_{i_{min}} \leftarrow$ Region connected to n_i with the minimum
        uncertainty value
    if $(\varepsilon_{n_i,n_{i_{min}}} < \varepsilon_{\lambda_{bhat}}) ==$ TRUE **then**
        $n_{new} \leftarrow$ Region resulting from merging
        n_i and $n_{i_{min}}$
        Update $[\mu^A_{\mathcal{X},n^*}; \mu^A_{\mathcal{Y},n^*}; \tilde{\sigma}^2_{A,n^*}; \tilde{\sigma}^2_{A,n^*(sum)}]$ *by*
        removing n_i *and* $n_{i_{min}}$ *data and adding* n_{new}
    else
        Eliminate n_i *data from* $\tilde{\sigma}^2_{A,n^*(sum)}$
    goto *loop* Until resulting regions cannot be merged
            among them anymore.

2.4 DBN representation

By taking the grown regions properties previously calculated as input data, this step generates a probabilistic inference architecture that facilitates the tracking of future agents. A DBN architecture is employed to represent the motion of observed agents in an environment. DBNs enable to include dependencies between involved random variables as time evolves. DBNs facilitate the representation of different inference levels related to agents' dynamics and incorporate the variables' uncertainties when predicting future instances. In this work, the lowest level of inference corresponds to measurements Z_k. States of agents, X_k, represent a medium inference level which captures continuous information of agents. Super-states A_k and n_k correspond to the top level of inference which consists of the complete activity that an agent performs A_k together with its respective discretization of regions $n_k \in n^*$; where n^* represents the set of large areas obtained from Algorithm 1. In such a top level, activities can be seen as a set of discrete sub-tasks executed

one after the other. Each sub-task is described as linear models which define the expected dynamics of agents according to their location in the environment.

The employed DBN architecture is depicted as in Figure 2.8. Each inference level is identified with a different color and arrows represent dependencies between variables. A dotted rectangle represents a single time instant k where the three levels are related to each other through conditional dependencies. The proposed DBN assumes that observations are continuous position values that can be modeled as Gaussian distributions. Similarly, agents' states are modeled as a multivariate normal distribution that carries information related to the positions and time derivatives of the agent in question.

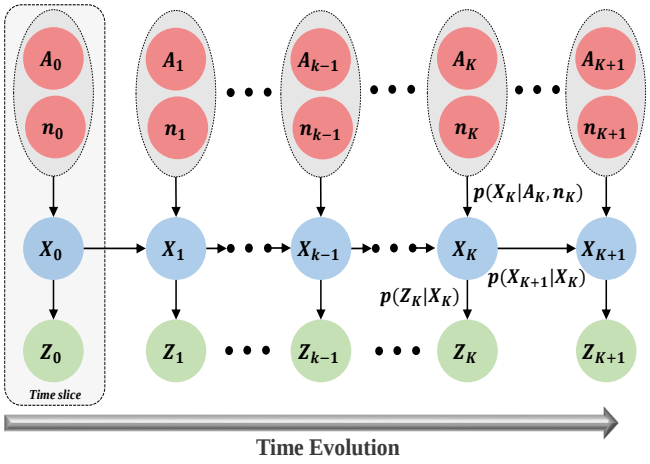


FIGURE 2.8: Proposed DBN architecture for modeling abnormalities.

From Figure 2.8, it is possible to see that each time slice of the proposed representation involves three conditional dependencies:

- $p(Z_k|X_k)$ which is the probability of obtaining an observation given the agent's state. The measurement model shown in equation 2.6 is used for making such inference.
- $p(X_{k+1}|X_k)$ represents the probability of obtaining a future agent's state given its present one. The dynamic model shown in equation 2.7 (see also equation 2.15) is used for making such inference.
- $p(X_k|A_k, n_k)$ expresses the probability of having the agent's state X_k given the super-state n_k related to the activity A_k.

Let $\boldsymbol{x}_{n_k}$ be the spatial components covered by the region n_k and $\dot{\boldsymbol{x}}_{n_k}$ be their correspondent quasi-constant velocity components. Note that mean values of the last two variables belong to the set of properties related to regions' spatial centroids and dynamical models calculated in Algorithm 1 such that $\bar{\boldsymbol{x}}_{n_k} \in \mu^A_{\mathcal{X},n^*}$ and $\bar{\dot{\boldsymbol{x}}}_{n_k} \in \mu^A_{\mathcal{Y},n^*}$.

Since n_k fixes a specific dynamical model, it is possible to approximate the control input in equation 2.15 as:

$$\hat{g}_{A_k}\Big(HX_{(l),k}\Big) \simeq \bar{\dot{\boldsymbol{x}}}_{n_k} \tag{2.24}$$

where it is assumed that $HX_{(l),k} \in \boldsymbol{x}_{n_k}$. As shown in equation 2.24, by knowing the agent's current region n_k, it is possible to approximate the function g_{A_k} as the region's mean velocity components of the proposed discretization process.

Proposed DBN depends on the previous state information for predicting future instances. Since the state of an agent (l) is composed of its positions and m time derivatives, such that $X_{(l)} = [\boldsymbol{x}_{(l)}\ \dot{\boldsymbol{x}}_{(l)} \cdots \boldsymbol{x}^{(m)}_{(l)}]^T$, by increasing the number of derivatives m, more information from the past is considered when making predictions. This work only considered the agents' velocity, i.e., $m = 1$, as part of the states. Such a choice assumes a sampling time that enables to capture the agents' motions and approximate them as piecewise constant velocity models. Our DBN can be seen as hierarchical structures containing both model selection and state estimation.

2.5 Abnormality detection

To perform abnormality detection based on probabilistic inferences, we set KFs to track agents' continuous states X_k based on models in regions n^*. Figure 2.9 summarizes the abnormality detection process for analyzing new unseen behaviors/maneuvers of agents in an environment.

Since states of the agents are composed of continuous variables whose dynamical and observation models are linear and their noise can be assumed as Gaussian distribution, this work considers a switching KF approach based on locally linear models previously obtained in Algorithm 1. As shown in Figure 2.10, proposed KFs are built based on identified regions' information.

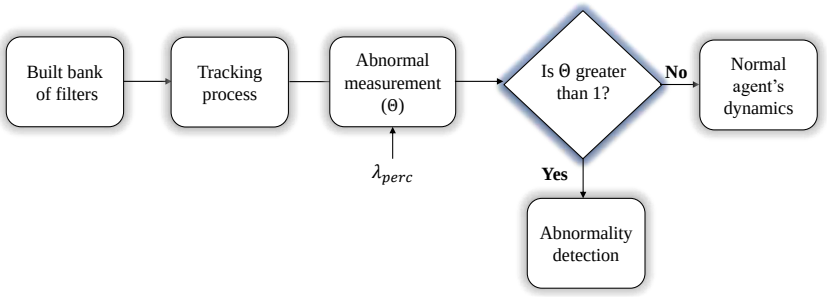


FIGURE 2.9: Proposed steps for detecting abnormalities.

As mentioned before, the proposed approach is based on local linear models (see in equation 2.15) where control inputs are modeled as shown in equation 2.24. KFs employ such models for predicting agents' future states. The error of such predictions can be used to build a normality indicator of observed agents' motions.

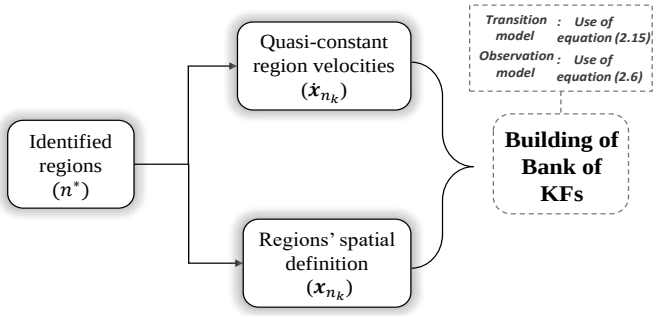


FIGURE 2.10: Proposed building of KFs for switching purposes.

KFs' error is defined as the innovations generated by all created (normal) filters valid in the current agent's location. As mentioned previously, each region's model can be seen as a multivariate Gaussian distribution defined by expected velocities, $\mu^A_{\mathcal{X},n}$ and their variances $\tilde{\sigma}^2_{A,n}$, where $n \in n^*$. By determining a percentage threshold $\lambda_{perc} \in [0, 1]$ that establishes the normality limits of observed dynamics, it is possible to obtain the vector $\Delta\tilde{Y}_{n(perc)}$ containing the maximum allowed deviations from expected dynamics $\mu^A_{\mathcal{X},n}$. Let θ_{n_k} be the KF's innovations divided by the maximum allowed deviations in the region $n_k \in n^*$, such that:

$$\theta_{n_k} = \frac{abs(\tilde{Y}_k)}{\Delta\tilde{Y}_{n(perc)}}. \tag{2.25}$$

Note that θ_{n_k} is a vector that includes the normalized innovations of the d components of the scene, and $\tilde{Y}_k$ is defined in Equation 2.10. The final abnormality

measure is defined as the maximum value of such vector:

$$\Theta_k = max(\theta_{n_k}). \tag{2.26}$$

Abnormal dynamics can be identified automatically by the proposed method when the abnormality measurement Θ_k is greater than 1; whereas the normal dynamics are inside the range $[0, 1]$. Observations detected as abnormal can be used to create ulterior linear models that can be added into the set of KFs. Such a process allows the system to learn new (abnormal) models incrementally and use them in future instances for prediction and tracking purposes. Figure 2.11 shows how ulterior KFs can be represented into a hierarchical scheme. Normal KFs employed for detecting abnormalities are indexed as b, whereas anomalies which can be potentially added into new KFs models are indexed as c.

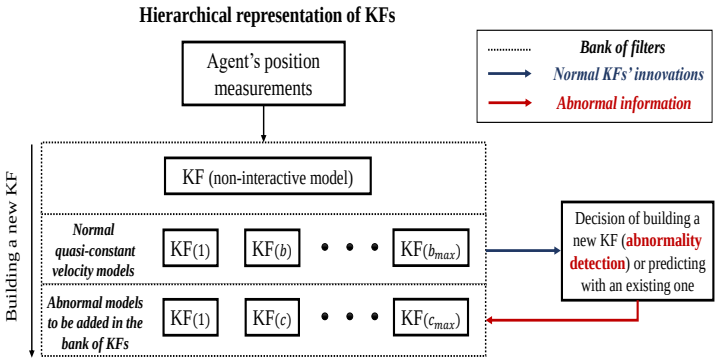


FIGURE 2.11: Scheme of abnormality detection.

Results of this work focus on the identification of abnormal situations that can be integrated into a set of KFs incrementally. In our approach, abnormalities are associated with the system's incapacity of explaining observations based on previously characterized models. The whole proposed method is tested with real measurements taken from a vehicle that performs diverse tasks in a closed environment. The following section describes the employed dataset in detail and the acquired results.

2.6 Experimental results

The proposed method for modeling, understanding and predicting dynamic relationships among moving agents and their external surrounding environment from exteroceptive information (positional information) is tested in scenarios. Accordingly, real and simulated data is taken into consideration for modeling agents' motions. Acquired results for both cases are presented in the following subsections.

2.6.1 Abnormality detection based on GP approach

A dataset based on a real vehicle that moves inside a closed environment is taken into consideration to test the proposed method for abnormality detection. Experiments and information about the vehicle are provided as follows.

2.6.1.1 Real dataset

First of all, it is relevant to mention that the following experiments were performed in collaboration with the Intelligent Systems Laboratory, Department of Systems Engineering and Automation of the University Carlos III de Madrid, Spain.

For testing the proposed algorithms in the detection of abnormalities, an initial simple task (defined as normal) executed by a vehicle inside a closed environment is considered. Additionally, different situations (considered as abnormal) where the vehicle deals with pedestrians while performing its initial task. Before describing each scenario here studied, it is necessary to explain the involved sensors and the real-time acquisition process of the vehicle.

The vehicle iCab [76] is equipped with a binocular camera Bumblebee 2 for capturing stereo environment information which provides color images of 640x480@20Hz. Moreover, it contains an installed Lidar Velodyne Puck VLP-16 to gather relevant information from its surroundings. Such sensor returns 16 laser scanner signals with a range of 360 degrees over the horizontal and 30 degrees over the vertical axes. Raw information from laser scanners is processed to compose point clouds at 10Hz. For the low-level control, the vehicle provides the actual steering angle and linear velocity based on the motor encoders at 20Hz.

FIGURE 2.12: Real iCab vehicle.

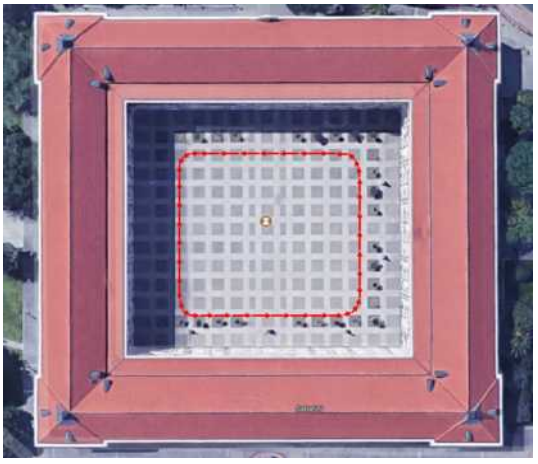


FIGURE 2.13: Normal dynamics in real environment structure.

Figure 2.12 shows the vehicle used for performing the proposed experiments. Additionally, the environment where the iCab vehicle moves is a closed rectangular square plaza part of the Sabatini building structure shown in Figure 2.13.

The vehicle's positions provided by the odometry modality are used as the sensory data in the work of this chapter. Such positions are mapped into Cartesian coordinates which represent the environment space where the vehicle moves. Accordingly, the Velodyne point cloud is processed to generate the x and y state space position that describes the vehicle's dynamics inside the scene. Obtained odometry data presents an accuracy around 10 cm. Additionally, an error of 0.5 degrees has to be considered due to the data acquisition process [77]. A separate computer is necessary to gather the final position outputs (at around 9Hz) due to a considerable computational cost of the point cloud processing.

All the synchronization processes are performed by a software prototyping tool called ROS [78]. Such tool is responsible for the communication between processes

and computers. From this viewpoint, the synchronization of the involved sensors is managed in a configurable layer which is transparent to the developers. Each experiment publishes the raw data with its respective timestamp values. Hence for post-processing and analysis, such timestamps are vital to computing correctly the streams of produced data used in this work.

The employed dataset uses standard ROS messages such as *LaserScan*, *PointCloud2*, *Odometry*, *Image*, *controls* and others. Such standardization of messages is essential for future compatibilities with other systems. The methodology for extracting and saving data is based on a tool in ROS called *rosbag*. Such tool saves sensory information in the form of standard messages with its respective timestamp such that produced data can be used for analysis and debug purposes.

The primary objective of the dataset is to create a collection of sensory data that emulates a perimeter monitoring scenario considered as the normal situation. As mentioned before, this work considers different scenarios based on anomaly vehicle's behaviors in the scene while performing the regular perimeter monitoring. Accordingly, three different scenarios are presented in this chapter consisting of normal activity and two types of deviations from it (abnormalities). Consequently, each scenario is explained in more details.

Scenario I (Perimeter Monitoring): The vehicle performs a rectangular path surrounding a closed environment (see the red trajectory in Figure 2.13) without any obstacles as shown in Figure 2.14.

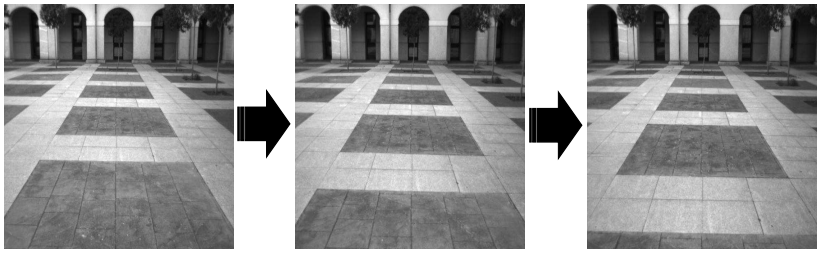


FIGURE 2.14: Frames of Perimeter monitoring maneuver from a first person perspective.

Scenario II (Avoidance Maneuver): While the vehicle executes a perimeter monitoring task, two static pedestrians are placed in different locations interfering with its path. In this scenario, the vehicle performs an avoidance maneuver to surpass the static pedestrian and continues the perimeter monitoring activity.

Figure 2.15 shows the temporal evolution of the avoidance maneuver from a first-person perspective. As can be seen, when the vehicle observes a static pedestrian, it surrounds him and then continues its trajectory.

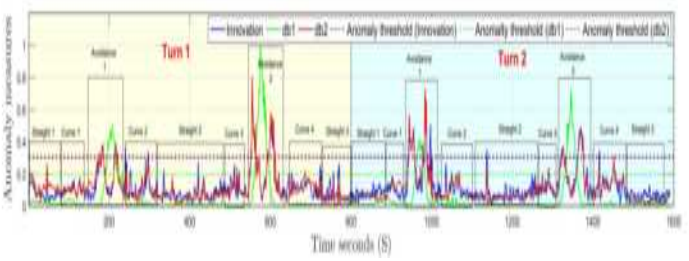


FIGURE 2.15: Frames of pedestrian avoidance maneuver from a first person perspective

Scenario III (Stop Maneuver): While the vehicle executes a perimeter monitoring task, it encounters in each lap two moving pedestrians that cross in front of its path. In such encounters, the vehicle's reaction consists of an emergency stop maneuver; then it continues its regular path as soon as the pedestrian leaves its field of view. A vehicle's first-person perspective of the temporal evolution of the stop maneuver is provided in Figure 2.16.

FIGURE 2.16: Frames of emergency stop maneuver from a first person perspective

2.6.1.2 Experiments

As mentioned before, three main scenarios are considered: *(i)* perimeter monitoring task *(ii)* Avoidance maneuver and *(iii)* Emergency stop maneuver. Unseen maneuvers, i.e., situations in *(ii)* and *(iii)* represent abnormalities from the regular perimeter monitoring task.

Experimental setup for learning normality

Displacements of the vehicle are extracted from innovations generated by the non-motivated KF (see equation 2.5). Data employed to analyze the normal perimeter control task is depicted in Figure 2.17.

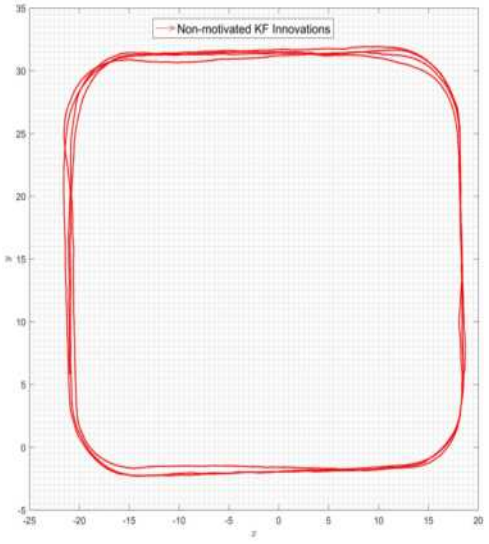
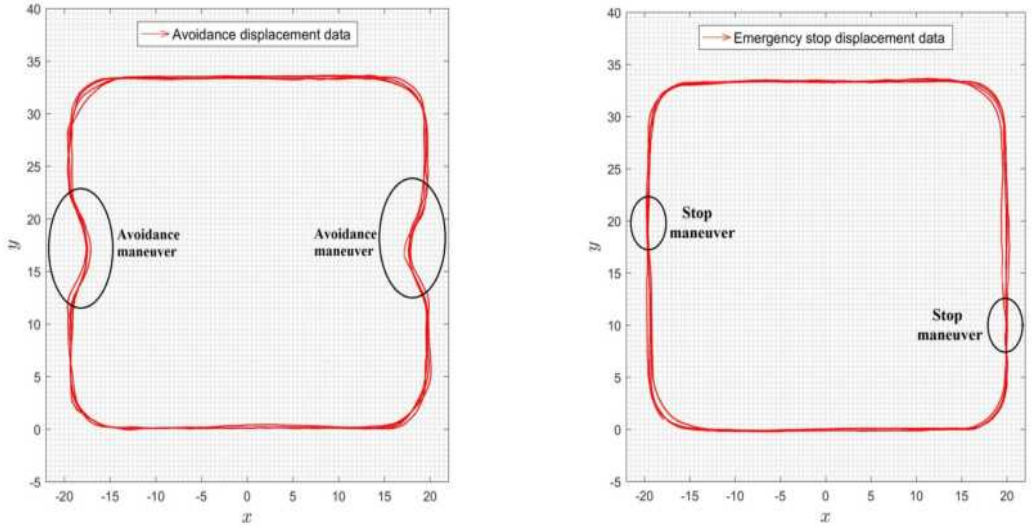


FIGURE 2.17: Displacement data for defining vehicle normal behavior

The latter information depicted in Figure 2.17 is used to define the environment normality and detect anomaly maneuvers introduced in Scenarios II and III. Displacement information from abnormal scenarios are shown respectively in Figures 2.18a and 2.18b.



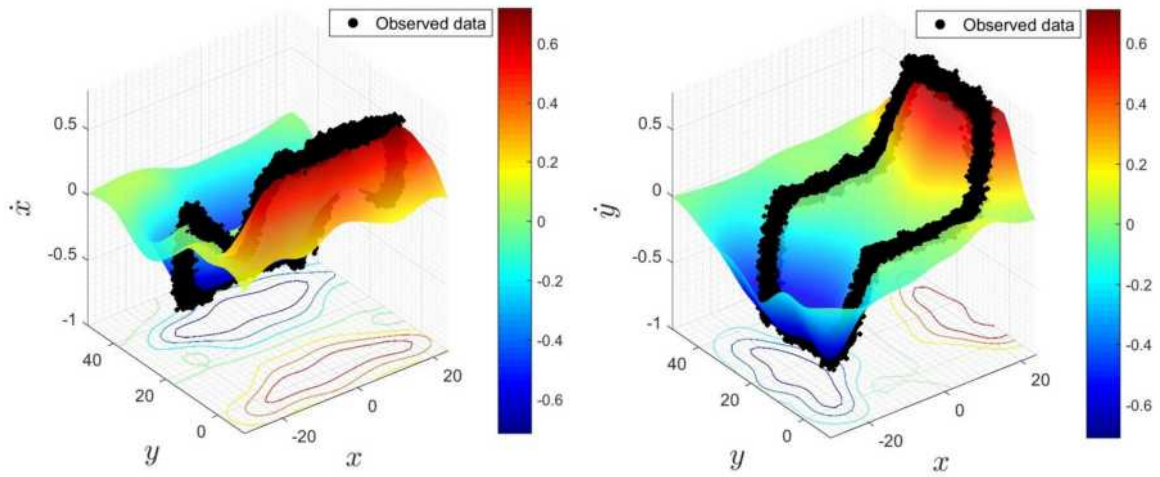
(A) Avoidance maneuver (B) Emergency stop maneuver

FIGURE 2.18: Displacement data used for testing abnormalities in vehicle behaviors.

Images shown in Figure 2.18 depict the abnormality cases in the trajectory data produced by the presence of pedestrians in the scene. Consequently, the whole information in both images is used to test the capability of the proposed methodology in recognizing abnormalities from the regular control monitoring task.

Threshold setting: The proposed method requires the setting of three threshold values, each of them already discussed in previously. For the experiments shown in this article, each of them is set as follows: *(i)* $\lambda_{val} = 0.7$ which selects the most certain grid points to be analyzed by the proposed method *(ii)* $\lambda_{bhat} = 0.7$ which facilitates the merging of similar neighbor OS regions and *(iii)* $\lambda_{perc} = 0.9$, which enables the recognition of abnormal motions based on deviations from previously learned models. It was observed that values higher than 0.5 do not reject relevant data for generating the DBN.

Based on trajectories that describe the perimeter monitoring activity, see the red path in Figure 2.17, it is applied a GP regression that follows the inputs/outputs of Figure 2.4. Innovation components $\dot{x}$ and $\dot{y}$, approximated by the GP and mapped into coordinate positions (x, y), are presented in Figures 2.19a and 2.19b respectively.



(A) GP over x data. (B) GP over y data.

FIGURE 2.19: GP approximation of vehicle dynamics over the environment.

Additionally, as proposed in equation 2.18, uncertainty values generated by innovation components are summed up to obtain an uncertainty measure in each environment location. A resulting surface containing coupled uncertainties of GP estimations onto the whole scene is shown in Figure 2.20.

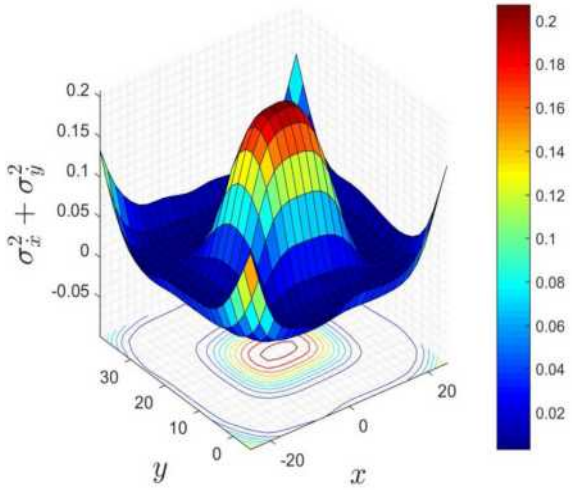


FIGURE 2.20: Joint variance (uncertainty measurement) produced by normal vehicle task: Perimeter control

The scheme presented in Figure 2.6 shows the coding of GP outputs into an RGB image. Accordingly, it is possible to identify environment locations where GP estimations have a high certainty. By doing that, it is possible to obtain an image that encodes the normal task dynamics in RG colors and uncertain locations are depicted in white. Such map is presented in Figure 2.21.

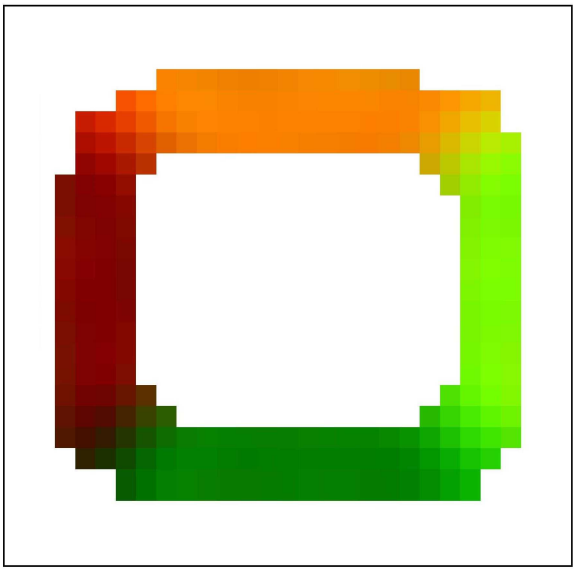
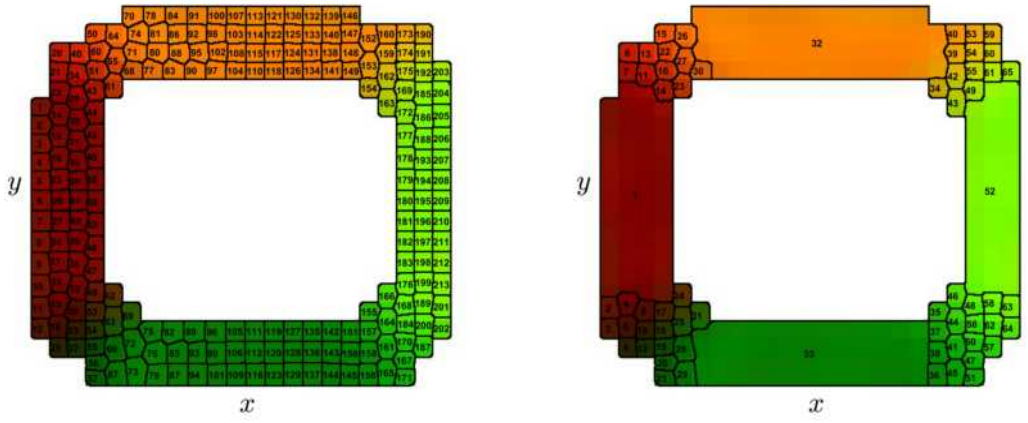


FIGURE 2.21: Image version of displacements approximated by a GP applied on perimeter control task data

As explained in section 2.3.3, a SP OS algorithm is applied over valid GP estimations, i.e., $\mathfrak{X}_{A,\lambda_{val}}$, $\mathfrak{Y}_{A,\lambda_{val}}$ and $\mathfrak{y}_{A,\lambda_{val}}$. Figure 2.22a shows the result of SP OS for the perimeter monitoring task. Subsequently, by applying the region growing

approach, see Algorithm 1, large regions where quasi-constant velocity models are valid can be obtained (see Figure 2.22b).



(A) Over-segmented GP version. (B) Grown regions.

FIGURE 2.22: Segmentations of GP perimeter control information into zones where quasi-constant velocity models are valid.

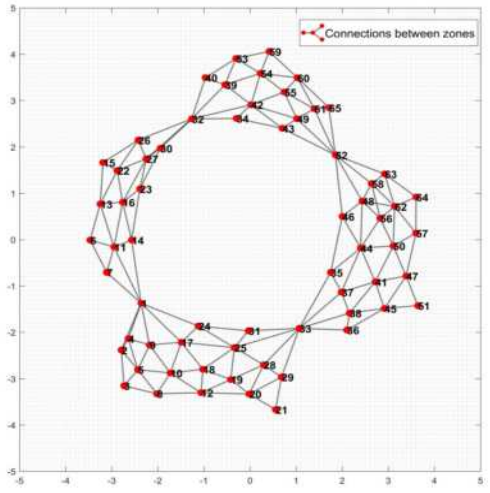
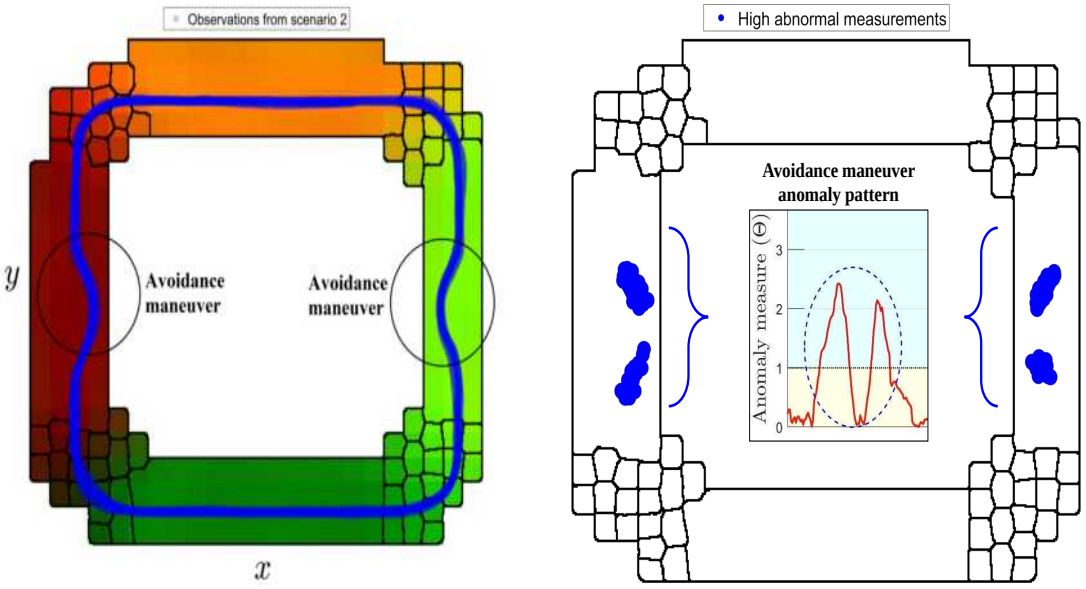


FIGURE 2.23: Graph associated to the final generated quasilinear dynamical zones based on the perimeter control activity.

Experimental setup for abnormality detection of unseen Maneuvers: Each zone that is shown in Figure 2.22b is used to create a KF valid in the spatial area in question. It is possible to represent all connections between produced zones by the graph presented in Figure 2.23. As can be seen in Figure 2.22b, 64 zones are obtained where quasi-constant dynamical models are valid. In other words, our proposed method decomposes the perimeter monitoring task into 64 KF motivated linear dynamical models extracted from GP valid data. Each linear dynamical model (see equations 2.15 and 2.24) is employed for prediction and abnormality detection purposes.

In this work, two different scenarios for abnormality detection are considered. They consist of unseen maneuvers due to interactions with pedestrians in the environment while the vehicle is performing the perimeter control task. From this viewpoint, the final objective of the proposed strategy is to detect and identify both types of unseen maneuvers based on the already characterized normal situation.

Static pedestrian avoidance: Figure 2.24a shows in blue the measured locations of a vehicle that performs two avoidance maneuvers during the perimeter monitoring task. The background colored image displays the identified regions where quasi-constant velocity models are valid based on the regular perimeter monitoring task.



(A) Avoidance maneuver data on normal GP map

(B) Identified anomalies

FIGURE 2.24: Observed data and spatial abnormality detection related to the avoidance maneuver while performing the control task perimeter.

By considering innovations generated by the set of KFs based on the perimeter monitoring task (normality), it is possible to identify abnormalities in new trajectory data that does not correspond to already learned models. As explained previously, high innovation values from KFs indicate the presence of anomalies in the scene. Since this work considers a 2-dimensional environment, two values of innovations are obtained at each time instant k. As shown in equation 2.25, the vector θ_{n_k} is obtained by taking the absolute value of innovations and normalizing them according to the maximum allowed deviations. The final anomaly measurement consists of the highest value of the vector θ_{n_k} (see equation 2.26).

Accordingly, Figure 2.25 presents the anomaly measure Θ_k through time obtained by applying the normal perimeter monitoring model to observations from scenario II.

Three main behaviors are recognized in the time series and presented in Figure 2.25. They correspond to "avoiding maneuver", "curve execution" and "straight path". As can be seen, parts of the avoidance maneuver and few points of the curve execution are detected as abnormal, i.e., $\Theta_k \geq 1$. Since avoidance maneuvers were not observed before, it is understandable that they produce high peaks of abnormality in points where the vehicle was supposed to go in a straight path. Curve points that present high anomaly values correspond to parts of the turns that do not assemble precisely with the maneuvers observed previously. Nonetheless, note that curves do not produce significantly abnormal measurements as avoidance maneuvers do.

It is relevant to mention that the anomaly pattern related to the avoidance maneuver, i.e., abnormal measurements inside blue ovals in Figure 2.25, is space independent. In other words, if there is an avoidance in any other large region of the environment (e.g., 1, 32, 33 and 52 shown in Figure 2.22b), a similar pattern will appear as it occurs. Additionally, note that straight path motions are identified clearly as normal behaviors concerning the standard perimeter control task. Figure 2.24 shows observations of scenario II that produced high abnormalities when using perimeter monitoring experiences for building the inference models. Two anomaly zones are obtained each time that an avoidance maneuver is performed. A single compact zone (in blue) is not formed due to the straight path in the avoidance maneuver which follows the regular perimeter monitoring task. Position and displacement information related to anomaly zones (shown in Figure 2.24) can be included in the current bank of filters as proposed in the scheme displayed in Figure 2.11.

Emergency stop maneuver: Figure 2.26a shows in blue the vehicle's locations while executing perimeter monitoring with two stop maneuvers. The colored

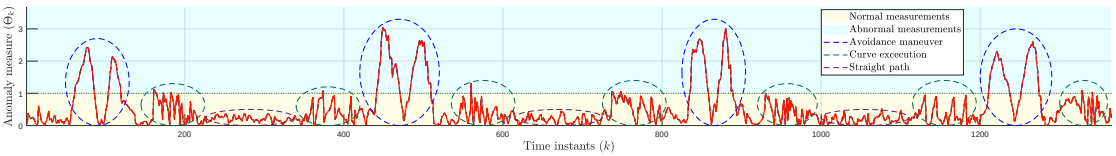


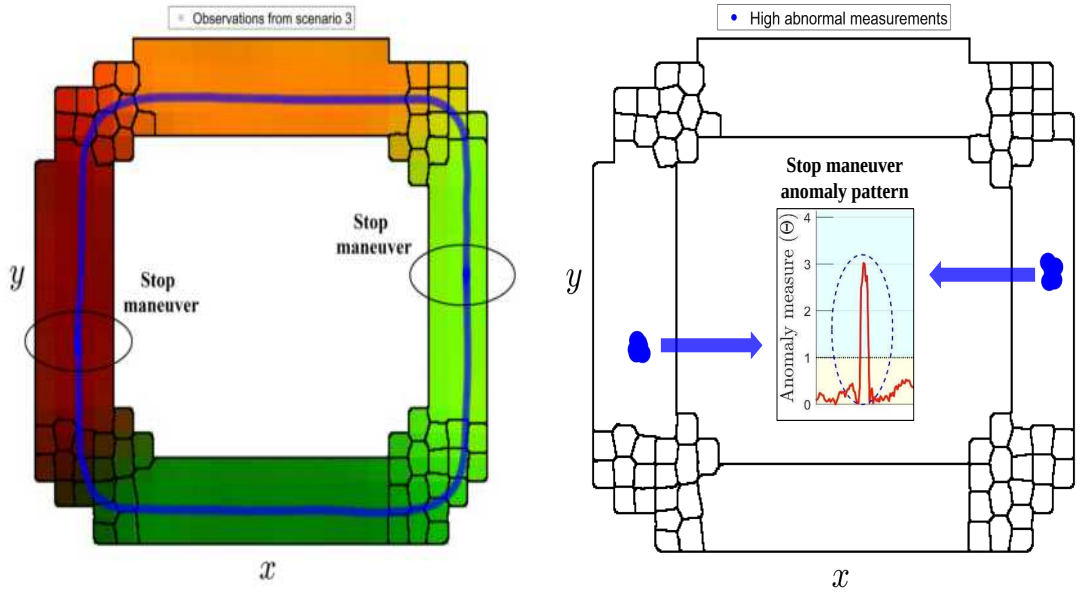
FIGURE 2.25: Abnormality measurements through time for perimeter control activity with avoidance of static pedestrians.

background contains the identified regions where quasi-constant velocity models are valid based on the perimeter monitoring task.

Similar to the results in the static pedestrian avoidance case, anomaly measurements generated by models trained with the perimeter monitoring task applied to the scenario III are shown in Figure 2.27. Three patterns can be distinguished in such image: "stop maneuver", "curve execution" and "straight path". It is clear the presence of an abnormality pattern (notice the blue ovals in Figure 2.27) consisting of a prominent peak that periodically shows up. Such peaks represent the emergency stop maneuver.

As explained previously in the avoidance case, subtle differences between curves performances produce some deviations from normality. Nonetheless, only a few curving points present abnormalities. Moreover, their level anomaly is low in comparison to the stop maneuver.

Done with scenarios I and II, Figure 2.26b shows the observations from scenario III where high abnormalities take place. Such measurements can be used potentially for creating new models into the set of KFs.



(A) Observed stop maneuver data on normal GP map.

(B) Abnormality zones recognized at analyzing data of scenario III.

FIGURE 2.26: Observed data and spatial abnormality detection related to the stop maneuver while performing the control task perimeter.

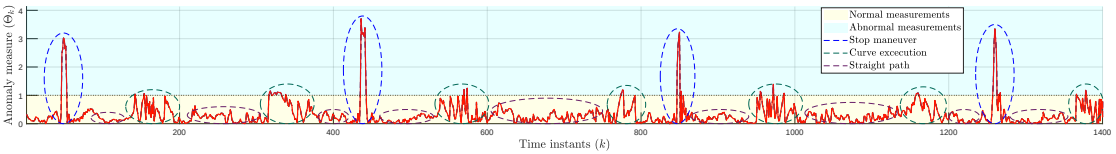


FIGURE 2.27: Abnormality measurements through time for perimeter control activity with emergency stop maneuver.

2.6.2 Classification of trajectories based on GP approach

We validate the proposed method with a simulated dataset introduced by [79]. Such dataset is composed of 19 labeled trajectory classes that move through a traffic intersection environment. Each trajectory class consists of 100 tracks destined for training and 500 trajectories designed for testing. As can be seen in Figure 2.28, trajectory data from the dataset is motivated by an automobile intersection environment where moving entities navigate. This dataset is employed to demonstrate our proposed methodology's accuracy in coding dynamics into zones that can be used for classification purposes. The dataset is composed of 2-dimensional spatial trajectories.

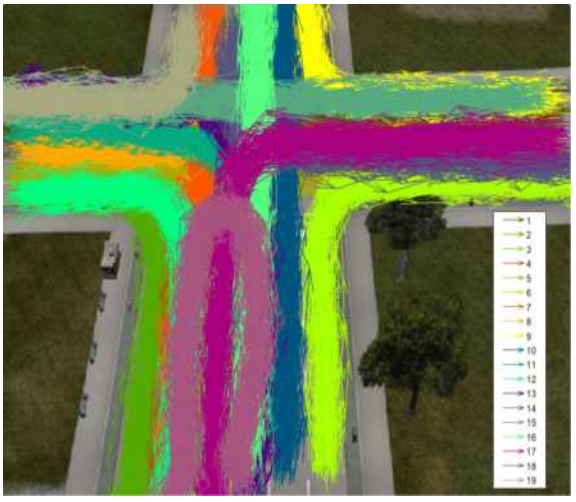
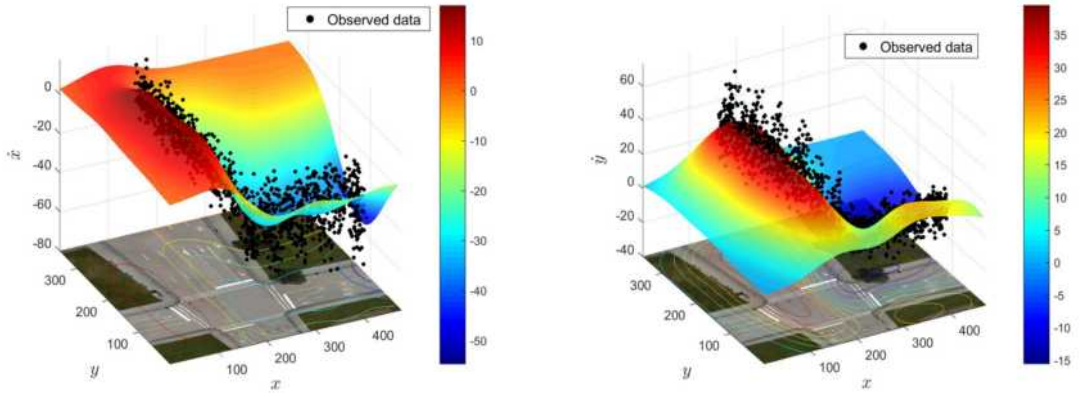


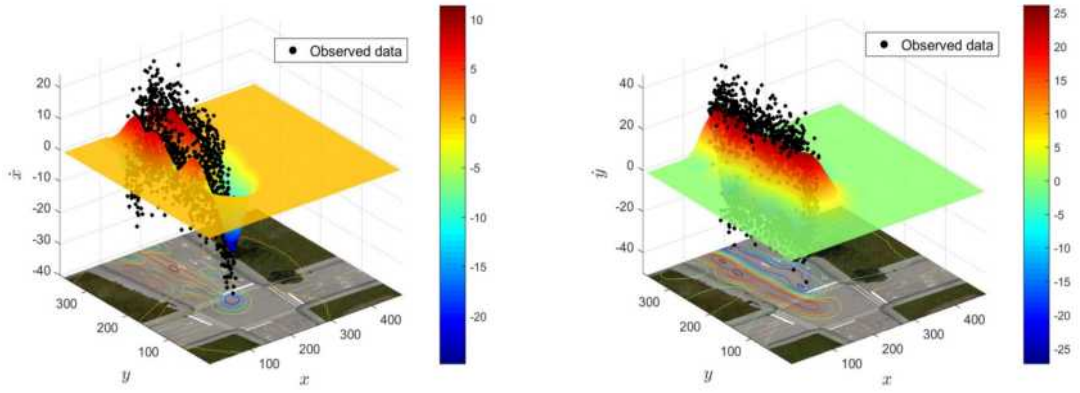
FIGURE 2.28: Intersection dataset layout.

The observations of the locations of moving agents are used to estimate agents' displacements such as proposed in Figure 2.5. Accordingly, a GP regression is executed for each activity (class) independently. The flow data $(\dot{x}, \dot{y})$ estimated by a GP approximation through the whole scene is shown in Figures 2.29 and 2.30 respectively for two trajectory classes. Such classes corresponding to a left turn activity (labeled as class 4) and a U-turn motion (labeled as class 18).



(A) $\dot{x}$ estimation (B) $\dot{y}$ estimation

FIGURE 2.29: GP results on traffic simulated data (class 4)

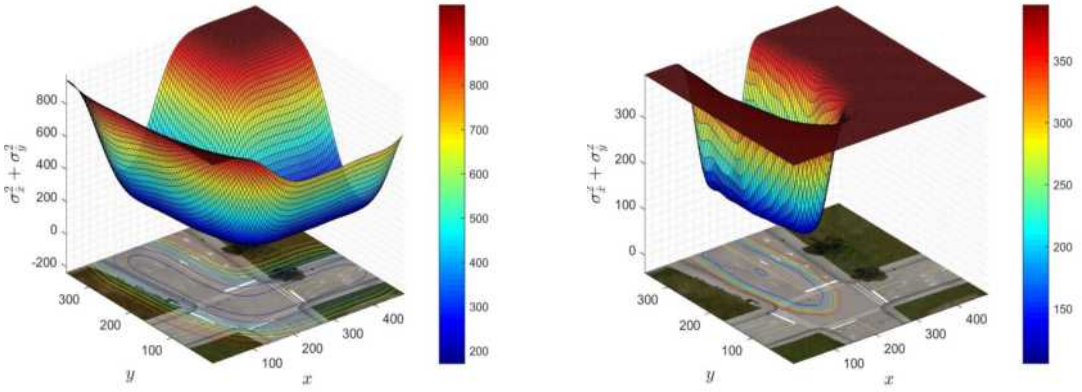


(A) $\dot{x}$ estimation. (B) $\dot{y}$ estimation.

FIGURE 2.30: GP results on traffic simulated data (class 18).

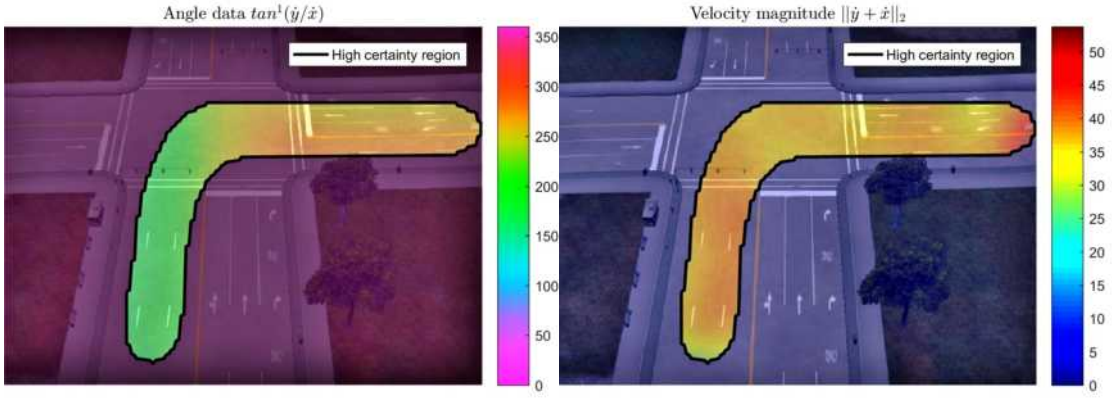
As discussed in section 2.1.2, a GP is composed of variables' mean estimations and a measure of uncertainty. Accordingly, mean approximated function is shown in both Figures 2.29 and 2.30, and the joint variance (see equation 2.18) of produced estimations are plotted in Figures 2.31a and 2.31b for classes 4 and 18 respectively.

As pointed out in section 2.3.1, it is possible to codify GP results in an image that represents the spatial information of the scene (see Figure 2.6). High certainty boundaries are used to cancel out information where not enough evidence that supports GP estimations. Figure 2.32 shows the images generated based on the GP results of activities 4 and 18 respectively.

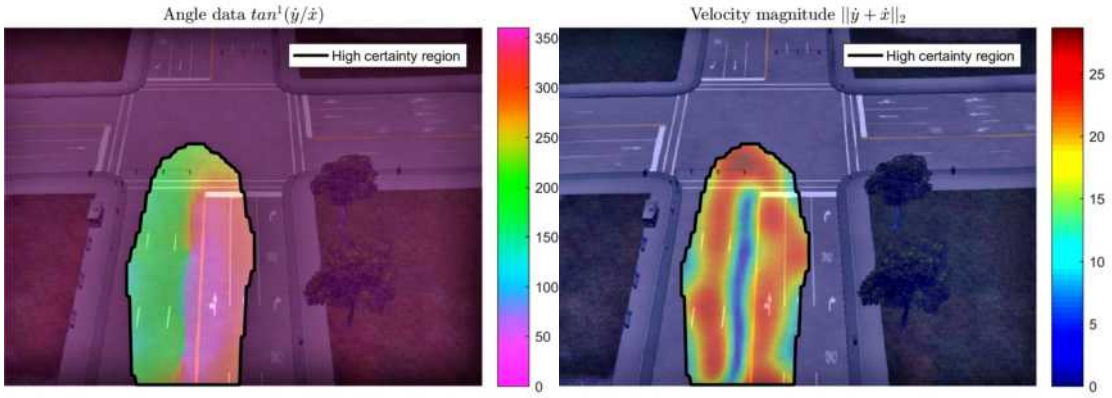


(A) GP Joint uncertainty associated to class 4. (B) GP Joint uncertainty associated to class 18.

FIGURE 2.31: GP joint uncertainty maps for two trajectory classes.



(A) Angle estimation class 4. (B) Speed estimation class 4.



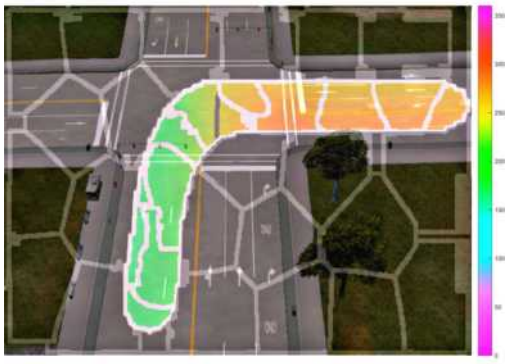
(C) Angle estimation class 18. (D) Angle estimation class 18.

FIGURE 2.32: Magnitude and angle estimations based on GP estimations applied to two different sets of trajectories.

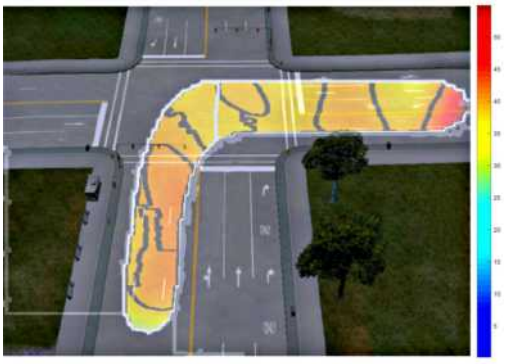
As can be seen from Figure 2.32, places where enough evidence support the GP estimations are bordered by a solid line that indicates the validity of GP approximated data.

The zones can be obtained where speeds and angles behave in a quasi-constant way. To obtain such zones, a SP approach see Section 2.3.3 is considered. Regions obtained by applying the superpixel algorithm to codified angle/magnitude images are shown in Figures 2.33.

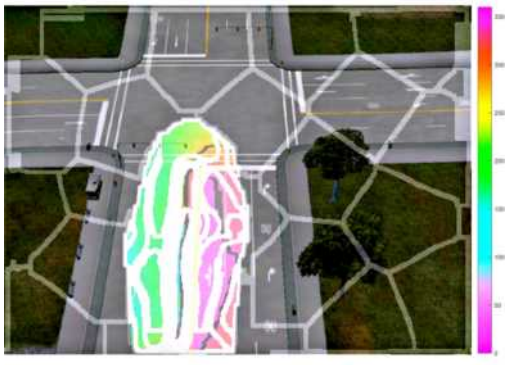
As can be observed in Figure 2.33, angle information requires less number superpixels to be fully codified into zones, e.g., in case of class 4, on Figure 2.33b, there are several zones limited by black border in order to completely codify speeds inside the map. In the other hand, Figure 2.33a shows that fewer black boundaries are necessary to divide the angle information. This result suggests that angles are more spatially stable than speeds.



(A) SP output for angle estimation class 4.



(B) SP output for Speed estimation class 4.



(C) P output for angle estimation class 18.

(D) SP output for Speed estimation class 18.

FIGURE 2.33: SP output based on GP estimations applied to two different sets of trajectories.

Trajectory classification: Assuming that C be the total number of classes. A certain amount of tracks is used as training set to obtain the zones where quasi linear dynamic models are valid. For validation, a separate group of tracks is used to measure the accuracy of generated zones at classifying unseen data by evaluating the similarity between the motion of a new trajectory and the previously characterized linear dynamics. To this end we consider a simple Euclidean distance between observed values of magnitude/direction and predictions based on linear models obtained during training.

Let $\epsilon_{mag,[c]}$ and $\epsilon_{dir,[c]}$ be the errors related to magnitude and direction by assuming that the trajectory belongs to class c. Additionally, we calculate an uncertainty measurement for each characterized class defined as:

$$\sigma^2_{joint,[c]} = \frac{\sum_{i=1}^{i_{total}} (\sigma^2_{i,joint})}{i_{total}} \tag{2.27}$$

where i represents the grid points of the zone (defined based on motions of class c) where the entity is,; and i_{total} is the total number of grid points inside the zone.

We concatenate the error and uncertainty information and obtain

$$\epsilon_{mag,[total]} = \{\epsilon_{mag,[1]}, \epsilon_{mag,[2]}, ..., \epsilon_{mag,[C]}\},$$

$$\epsilon_{dir,[total]} = \{\epsilon_{dir,[1]}, \epsilon_{dir,[2]}, ..., \epsilon_{dir,[C]}\},$$

$$\sigma^2_{joint,[total]} = \{\sigma^2_{joint,[1]}, \sigma^2_{joint,[2]}, ..., \sigma^2_{joint,[C]}\}.$$

By considering $\tilde{(\cdot)}$ as an operator that orders data in an ascending order and $id_{[c]}(\cdot)$ as a function that returns the index where information related to the class c is placed inside a set of data, it is possible to define a voting score of each class as:

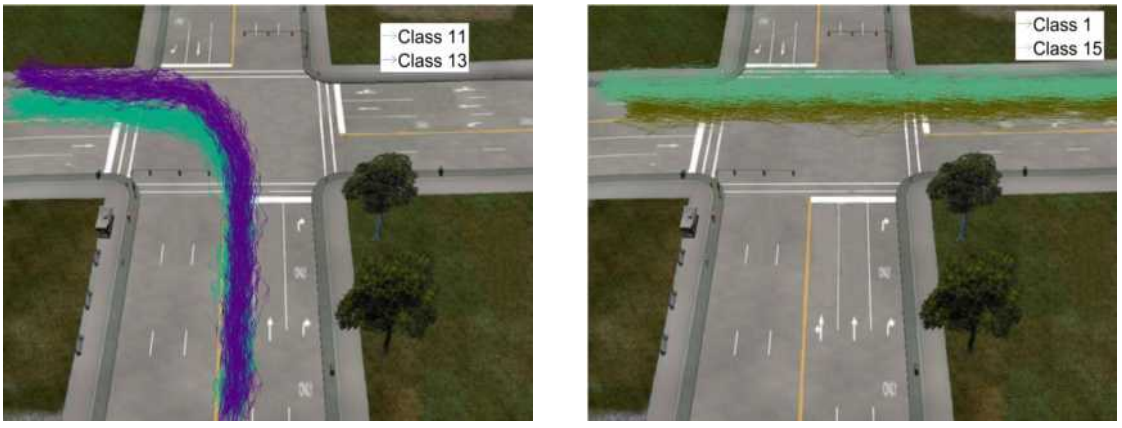
$$\begin{aligned} vote_{[c]} =& \alpha(id_{[c]}(\tilde{\epsilon}_{mag,[total]})) + \beta(id_{[c]}(\tilde{\epsilon}_{dir,[total]})) \\ &+ \gamma(id_{[c]}(\tilde{\sigma}^2_{joint,[total]})) \end{aligned} \tag{2.28}$$

where α, β, γ represent weights for magnitude, direction and uncertainty, respectively, for the voting process. A high weight indicates that the corresponding variable affects more the classification decision. We use the constraint $\alpha+\beta+\gamma = 1$.

Voting scores of each trajectory class are concatenated and the minimum value among them is chosen as the predicted class as expressed in equation 2.29.

$$class = \min_{i=1,...,C}(vote_{[i]}). \tag{2.29}$$

As mentioned previously, obtained maps are used for classification purposes by considering the dataset proposed by [79]. Following the voting process explained above, parameters in equation 2.28 are fixed as: $\alpha = 0.5$ $\beta = 0.2$ and $\gamma = 0.3$. The high importance assigned to the angle data is due to its observed spatial stability (see previous section). The overall classification rate of the proposed method over testing data is 87.59%. The confusion matrix for all trajectory classes is shown in Table 2.2.



(A) Training data from classes 11 and 13. (B) Training data from classes 1 and 15.

FIGURE 2.34: High confusion cases on traffic simulated data classification.

As can be seen from Table 2.2, there are two critical confusion cases where classes are present a misclassification percentage over 35%, it is the case of confusing classes 13 with 11 and 15 with 1. Accordingly, training data from the couple of classes 11-13 and 1-15 are shown in Figures 2.34a and 2.34b respectively. From both images, it is possible to see that the motion of individuals is very similar. The only thing that differs between classes in question is a subtle spatial separation because of lanes that are alongside each other.

Classes	Classes 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	77.2														22.8				
2		100																	
3			97.6				1.6				0.2						0.6		
4				100															
5					99.8							0.2							
6						100													
7			7.2				92.8												
8					0.2			99.8											
9	8								81.4						10.6				
10										94.6	0.2					5.2			
11											79		21						
12		7										93							
13											36.4		63.6						
14														100					
15	39														61				
16			15							8.4	1.6		3.4			71.6			
17				34													66		
18																		100	
19	0.6										0.6		1		11				86.8

TABLE 2.2: Confusion matrix for the trajectory classes of the traffic intersection dataset (values are in percentages). Empty cells represent zeros.

2.6.3 Discussions

This chapter has presented a single modality awareness model related to the position information. As pointed previously, this chapter presents a method for detecting abnormalities in observed trajectories by using a set of KFs built incrementally as occurring experiences. Such bank of filters is created based on a non-motivated dynamical model from which more complex dynamics are approximated and added into the available inference models.

A decomposition of GP regression based on a SP-like approach is applied to obtain a set of zones where quasi-constant velocity models are valid. Generated zones are employed for prediction and abnormality detection in real data from a vehicle performing a series of tasks in a controlled scene.

A strategy for detecting abnormalities based on innovation measurements is tested by using a real vehicle's trajectories. In this scenario, the proposed method used to analyze motion maneuvers of the agent while pedestrians are presented. Results suggest that our methodology facilitates the way of finding abnormalities in an online way and identifying anomaly observations (unknown maneuvers) that can be potentially learned as new models to be integrated into the set of KFs.

Since abnormalities in vehicle-pedestrian interaction cases can be automatically recognized and characterized, the present work can be used for understanding more complex scenarios for autonomous vehicles. Consequently, such information can be used for increasing the awareness of autonomous systems by understanding the vehicles' dynamics through multi sensors data.

Chapter 3

Multi-Sensorial Data for Learning a Multi-Modal Awareness System

An AA is like a human because it perceives the world and itself by using multiple sensors (senses). Accordingly, it is possible to use this set of information to give the agent some level of awareness about itself and its surroundings. In chapter 2, one modality is used for learning an awareness model. Instead, this chapter focuses on novel approaches to learn a multi-modal SA models for abnormality detection for a moving agent based on multiple sensors that observe the same phenomenon from different perspectives. Two different approaches are presented in this chapter: Semi-supervised GP-based and Unsupervised incremental learning approaches.

3.1 Semi-supervised GP-based approach

We introduce two layers of SA to increase the awareness of an agent: Shared Layer (SL) and Private Layer (PL). Besides, in this section we propose a method for abnormality detection based on multiple sensors that observe the same phenomenon from different perspectives.

Abnormalities can be first detected as deviations from Environment Centered (EC) models, i.e., from an observer viewpoint which does not have access to internal agent variables. Such a layer can be defined as a SL of SA since the observed

information, e.g., observed position and velocity can be measured easily from an external observer.

An observed agent can also have further information corresponding to what it can observe from a First Person Viewpoint (FPV) while a task is performed. Abnormalities related to unexpected observations acquired while performing a task can be considered as the essential information to define a PL of SA. Such information is available only to the agent itself. Accordingly, an external observer cannot access to such information and has to rely solely on SL information.

In order to do so, we aimed to extend the work performed with the vehicle (previously discussed in chapter 2). The results obtained in such chapter are based on positional information that can be observed from an external viewpoint. Hence, such information is considered as a SL data. As explained in chapter 2, the proposed method can generate locally uniform motion models by dividing a GP that approximates agents' displacements on the scene as shown in Figure 2.22b and provides a SL SA based on EC models. After that, such models are used to train image information taken from the vehicle's perspective in a semi-unsupervised way a set of GANs that produce an estimation of external and internal parameters of moving agents.

3.1.1 Private-layer SA modeling

In order to model the PL of SA, a GANs [80] are proposed to learn the normality pattern of the observed scene. Hence, it is used to learn the relationships among frames and their optical flow. GANs are deep networks commonly used to generate data (e.g., images) and are trained using only unsupervised data. The supervisory information in a GAN is indirectly provided by an adversarial game between two independent networks: a generator (G) and a discriminator (D). During training, G generates new data and D tries to understand whether its input is real (i.e., it is a training image) or produced by G. The competition between G and D is helpful for boosting the ability of both G and D.

Two channels are employed to learn the normality of the observed scene: appearance (i.e., raw-pixels) and motion (optical flow images) for two cross-channel tasks. In the first task, optical-flow images are generated from the original frames. In the second task, appearance information is estimated from an optical flow image.

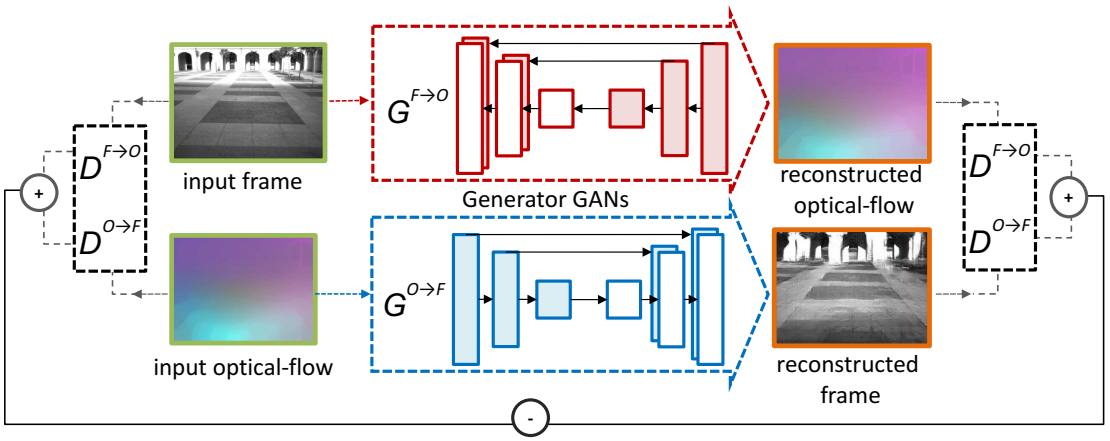


FIGURE 3.1: The two GANs structure: (i) $\mathcal{N}^{F\to O}$ generates optical-flow from frames by $G^{F\to O}$, and (ii) $\mathcal{N}^{O\to F}$generates frames from optical-flow by $G^{O\to F}$. Following with the corresponding Discriminators $D^{F\to O}$, and $G^{O\to F}$.

Let F_t be the tth frame of a training video and O^{th} the optical flow obtained using F_t and F_{t+1}. Remark: O_t is computed using [81].

Figure 3.1 shows two networks: $\mathcal{N}^{F\to O}$ which is trained to generate optical-flow from frames (task 1), and $\mathcal{N}^{O\to F}$ which generates frames from optical-flow (task 2). In both cases, inspired by [82, 83], here our networks are composed of a conditional generator G and a conditional discriminator D. G takes as an image x and a noise vector z (drawn from a noise distribution $\mathcal{Z}$) as inputs and an image $r = G(x, z)$ of the same dimensions of x but represented in a different channel as outputs.

Both G and D are fully-convolutional networks composed of convolutional, batch-normalization layers and ReLU nonlinearities. In case of G, we adopt the U-Net architecture [82] which is an encoder-decoder. D is proposed to be a *PatchGAN* discriminator [82] which is based on a "small" fully-convolutional discriminator $\hat{D}$. Additional details about the training procedure can be found in [82, 83]. During training, the output of $\hat{D}$ is averaged over all the grid positions such that final score of D is obtained with respect to the input. For testing purposes, we directly use the averaged scores of $\hat{D}$ as a "detector" which is run over the grid to detect the abnormality from the input frame (see section 3.1.1.1).

It is important to highlight that both $\{F_t\}$ and $\{O_t\}$ are here collected by using only the frames from *normal* scenarios (control perimeter activity) in the identified zones provided by GP (shown in Figure 2.22b). Accordingly, the absence of abnormal events at the training phase makes it possible to train the discriminators corresponding to our two tasks without the need of supervised training

data: G acts as an implicit supervision for D. We hypothesize that the latter lies outside the discriminator's decision boundaries because they represent situations never observed during training and hence treated by D as outliers. We use a *Bank of Discriminators* based on the identified zones provided by GP in Figure 2.22b, which is grouped into two sets: $Set1$ is trained on a straight path, and $Set2$ is trained over the curves as shown in Figure 3.2. The discriminator's learned decision boundaries can be used to detect unseen events.

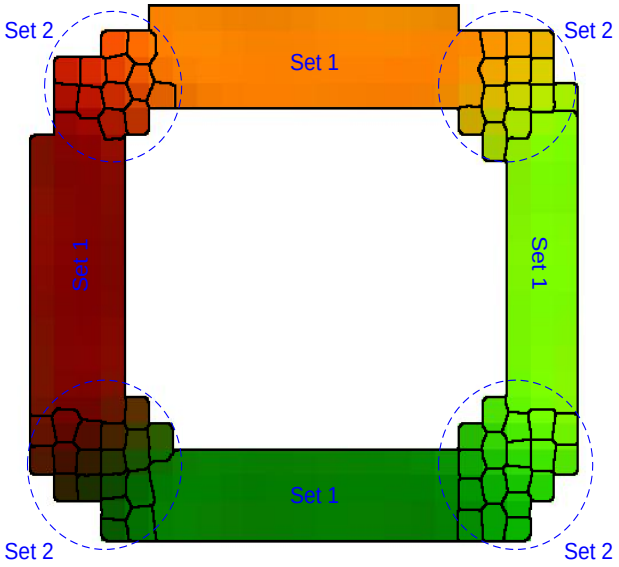


FIGURE 3.2: Spatial information in terms of zones used to train PL data.

3.1.1.1 Anomaly detection

Discriminators are used at the testing phase. More specifically, let $\hat{D}^{F\to O}$ and $\hat{D}^{O\to F}$ be the patch-based discriminators trained using the two channel transformation tasks (see Figure 3.1). Given a test frame F and its corresponding optical-flow image O, we first produce the reconstructed p_O and p_F using $G^{F\to O}$ and $G^{O\to F}$ respectively. Then, the pairs of patch-based discriminators $\hat{D}^{F\to O}$ and $\hat{D}^{O\to F}$ are applied respectively to the first and second tasks. Such operation results in two scores for the ground truth observation: S^O and S^F and two scores for the prediction (reconstructed data): S^{p_O} and S^{p_F}. The two scores are summed: $S_{observation} = S^O + S^F$, $S_{prediction} = S^{p_O} + S^{p_F}$. Besides, the values in $S_{observation}$ and $S_{prediction}$ are normalized into the range $[0, 1]$. Note that we do not need to produce the reconstruction images to use the discriminators. For instance, for a given position on the grid, $\hat{D}^{F\to O}$ takes as input a patch p_F on F and a corresponding patch

p_O on O. Moreover, a possible abnormality in the observation (e.g., an unusual object/movement) corresponds to an outlier with respect to the data distribution learned by $\hat{D}^{F\to O}$ and $\hat{D}^{O\to F}$ during training. The presence of the anomaly results in a low value of $\hat{D}^{F\to O}(p_F, p_O)$ and $\hat{D}^{O\to F}(p_O, p_F)$ (prediction) but a high value of $\hat{D}^{F\to O}(F, O)$ and $\hat{D}^{O\to F}(O, F)$ (observation).

Hence, in order to decide whether an observation is abnormal with respect to the scores from the current bank of discriminators, we simply measure the distance between predicted scores and observation scores starting from equation 3.1.

$$\tilde{Y} = S_{observation} - S_{prediction} \tag{3.1}$$

Furthermore, an error threshold $\tilde{Y}_{thres}$ is defined to detect the abnormal events: when $\tilde{Y}$ exceeds such threshold, the current agent's measurement is considered as an abnormal situation.

Experimental setup for abnormality detection

The proposed method is validated with data acquired from real vehicle during a perimeter monitoring task as described in section 2.6.1.1. Accordingly, captured video footage from a first person vision acquired with a built-in camera of the vehicle are used to test the PL.

As discussed previously, the bank of GANs are trained on the subsets of data based on GP zones. Here, the bank of GANs is composed of two major subsets: $Set1$ and $Set2$ (see Figure 3.2). Accordingly, each GAN detects the abnormality in the corresponding set on which it is trained.

Figures 3.3 and 3.4 compare the results obtained trough external viewpoint (SL data) and FPV images (PL data). Both figures show the time series of anomaly measurements based on pedestrian avoidance scenario. Anomaly detection results associated to the PL using the proposed bank of GANs are shown in Figure 3.4. Figure 3.4 shows three signals: The green and blue signals respectively show the computed signals by our GAN_1 (trained on $Set1$) and GAN_2 (trained on $Set2$). The red signal indicates the final abnormality measurement which is defined as the minimum value of GAN_1 and GAN_2. As it was expected, the obtained abnormality measurement in PL matches the SL results shown in Figure 3.3.

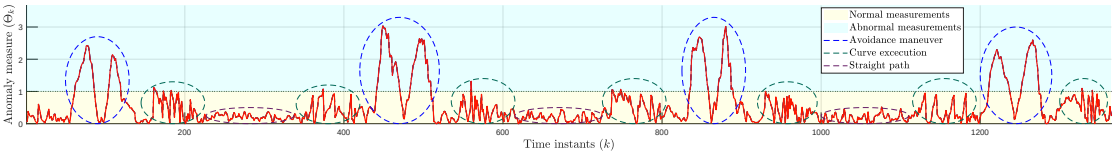


FIGURE 3.3: SL anomaly measurements: perimeter control activity by GP through time with avoidance of static pedestrians.

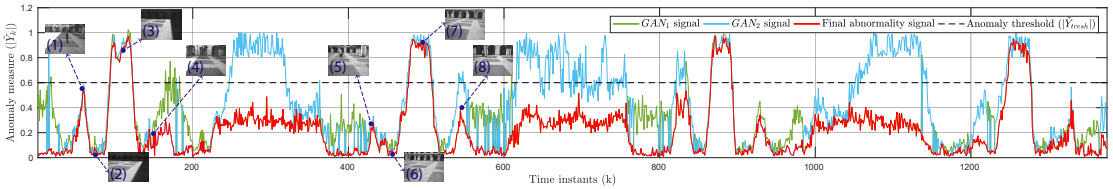


FIGURE 3.4: PL anomaly measurements: the distances between the observations and predictions by GANs during the time.

Different parts of the curve can be associated and explained by considering the corresponding images acquired from the on-board sensor. Specifically, the small peak identified with number 1 can be justified by the presence of the pedestrian in the field of view of the camera: the vehicle has not started the avoidance maneuver yet, thus it can be seen as a pre-alarm. The small peak in 1 corresponds to peak in 5. The latter is smaller due to the posture of the pedestrian (see correspondent images 1 and 7). The areas of the curve identified with numbers 2 and 3 or 6 and 7 correspond to the starting point of the abnormal maneuver and the avoiding behavior itself. It can be observed that peaks 3 and 7 are higher than the selected threshold and then correspond to an anomaly. After the small peak 4, that corresponds to the closing part of the avoidance turn, the vehicle goes back to the standard behavior. In particular, at this point of the curve, the vehicle is actually turning. In the wider area (from 220 to 380 secs.), the 'iCab' is moving straight. The slightly higher level of the abnormality curve in straight areas can be explained by a noise related to the vibration of the on-board camera due to the fast movement of the vehicle when increasing its speed.

It is notable that the signal generated by GAN_1 becomes higher in the curving areas since it is only trained on $Set1$ for detecting straight paths. Similarly, the GAN_2 which is trained on $Set2$ generates higher scores on the straight path. However, both GAN_1 and GAN_2 can detect the abnormality area (pedestrian avoidance) where both generate a high abnormality score.

Discussion

In this section, a multi-layer SA modeling is proposed to allow an agent to perceive the situations through different sensory modalities. Additionally, models of different SA layers can be integrated to build up a structured multi-modal self-aware behavior for an agent. As shown in section 3.1.1.1, the PL and SL layers provide complementary information regarding the situation awareness. Also, the advantage of introducing PL is to improve the awareness of the agent, for example abnormality anticipation.

However, the major problem of this method is the limitation on modeling different dynamics appearing on the same state-space coordinate (e.g. distinguish different displacement in the same point of the scene) which is due to the GP nature. Hence, we need to use another approach to tackle this issue in order to make it possible to represent different dynamics through a switching model.

In addition, up to now, the proposed approach considers observable data regarding a situation where an agent is involved. Nonetheless, internal variables of individuals (control parameters) have not been developed yet. Hence, in the following, we will also introduce agent's self variables.

Furthermore, a real AA, by its nature, needs to interact with a continuous dynamic permanent changing environment and continuously learns the novel unseen concepts (new concepts). In other words, the agent by itself has to understand what the new concepts are and then adapt itself (e.g. detect the new situation, learn a new model and update the current knowledge). For that reason consequently, in the following section an incremental learning procedure is presented.

3.2 Unsupervised incremental learning approach of switching models

This section highlights the idea of introducing an incremental learning process of dynamic models from data acquired along with the agent in the new experiences which in return facilitates the building of SA models. We propose an incremental adaptive process that allows the agent to learn a switching DBN model from recorded data. Such DBN models not only can predict (i.e., to generate) the new observed situations, but also they are capable of adaptively estimating the current

states by filtering data with respect to the most fitting model (i.e., to discriminate). In other words, the learned DBN models allow both prediction and estimation of situations different from the reference dynamic equilibrium (i.e, previously learned models) and then learning new/unseen concepts in an incremental fashion.

In the section 3.1, PL and SL are defined to learn an SA model for an autonomous agent. In this section, we include a new layer of SA that is related to agent's self variables, i.e., how own actions (changes in actuators) generate changes in the internal perception of own states. Such information consists of the controls of the agent's motion, i.e., steering angle and rotors' velocity. Note that such data also is considered as a private layer information but related to the internal state of the agent. We define this new layer related to the control parameter as Control Layer (CL).

3.2.1 Generic incremental learning structure

Each layer of SA includes different modalities. For each of them, a unified incremental learning procedure is designed (see Figure 3.5). The learning process is considered as a differential incremental process adding generative and discriminated knowledge to a reference model that describes a general dynamic equilibrium situation between the agent and the environment. Statistically significant deviations from such a dynamic equilibrium are recognized as abnormal situations whose characteristics are captured by the new learned models. Next to that, an agent can take advantage of new experiences when the reference situation of dynamic equilibrium is perturbed by adding new models integrated into the models that compose the dynamic equilibrium. A probabilistic framework based on a set of switching dynamical models is used to learn the SL and CL filters models incrementally. An incremental bank of cross-modal GANs is used to learn the PL models.

Figure 3.5 shows a flow chart that describes our proposed adaptive learning process. The latter allows agents to incrementally learn the additional knowledge necessary to describe a new observed situation. By encoding such knowledge into the agent's conditions of equilibrium trough probabilistic switching models, the idea of increasing the awareness of the agent becomes possible.

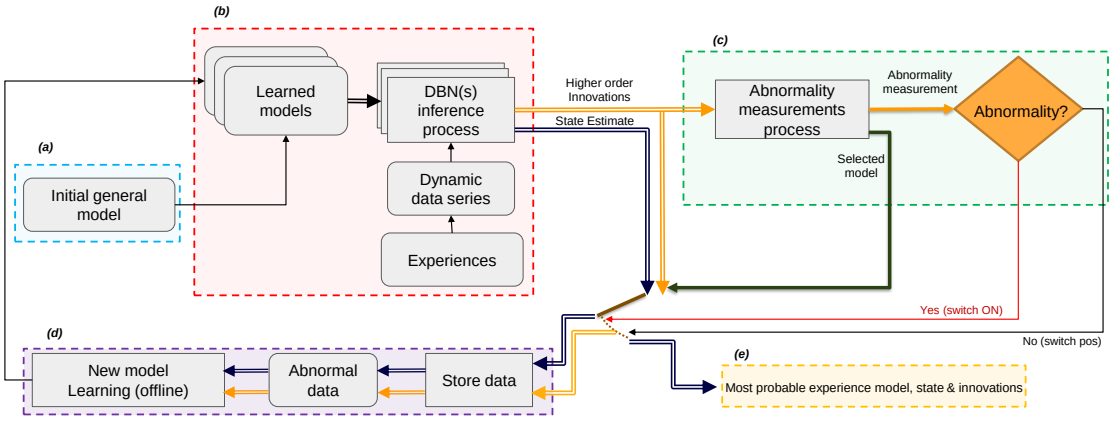


FIGURE 3.5: Generic Block Diagram of Incremental Learning process.

The learning of a new switching model starts by observing dynamic data series related to a given experience (Figure 3.5-b). Such data is filtered by using the initial reference model and by keeping track of deviations w.r.t the associated dynamic model (refer to Figure 3.5-a). The reference model consists of a simple filter whose dynamic model describes a basic dynamic equilibrium condition.

Accordingly, in the case of SL and CL, where low dimensional data like (position and velocity) and (steering angle and rotors velocity) are considered, the initial filter corresponds to a KF that assumes that the agent remains static (null speed). Such a filter makes reasonable predictions if there are no forces that affect the state of the agent, i.e., in the case where the agent does not interact with its surroundings. In that case, agent motions are only due to random noise oscillations of the state. In the shared level of SA position observations, noises can be produced either by the agent or external entities producing a set of noisy data series of sparse measurements. We define such filter as the UMKF as discussed in section 2.2.1. Such a reference filter is illustrated in Figures 3.6-a.1 and 3.6-a.2.

When UMKF applied along with a motivated experience, it produces a set of errors due to the fact that the agent follows a certain motivation modeled as surroundings' forces acting on it.

In the case of the private layer of SA, a pre-trained reference GAN is used as an initial general model which encodes an experience where the agent is moving straight on a clear path (see Figure 3.6-a.3). Similar to the UMKF general model, the reference GAN assumes a dynamic equilibrium existing between the environment and the agent but on a different modality (*i.e.*, first-person video data). In this case, the condition of equilibrium corresponds to the agent's visual data as it

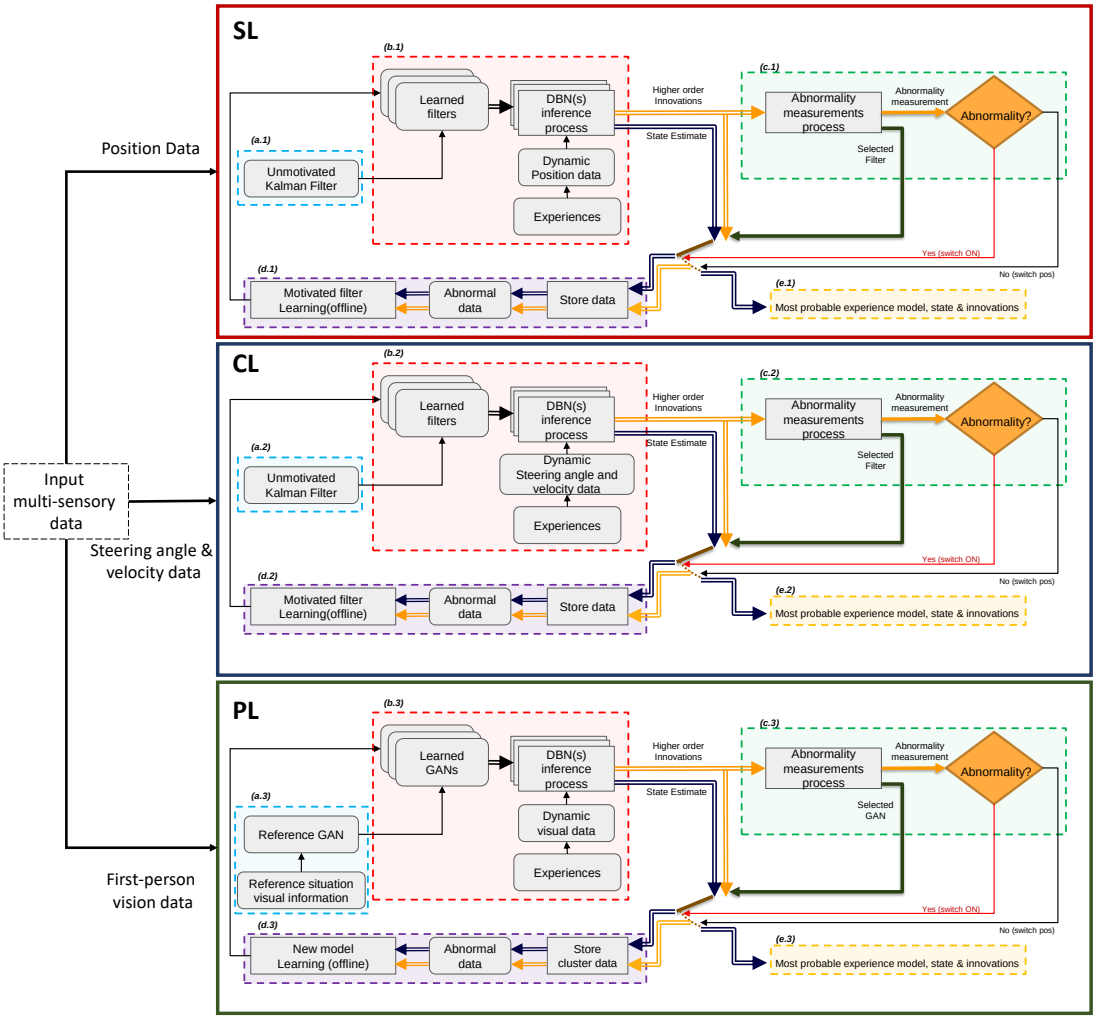


FIGURE 3.6: Learning Process in SL, CL and PL: for each experience the position information input to the SL, control information are employed for CL while the first person vision data are used as a source of information for PL. All layers structured based on the generic incremental adaptive training process.

moves linearly towards a point in the environment. Such a point can be seen as a stationary center of force that attracts the agent linearly. In case other forces were present in the agents' surroundings, the dynamics of visual data would change with respect to the one experienced with the straight movements, and consequently, a set of prediction mismatches between the GAN filter predictions and updating new observations would be produced as errors.

Considering the set of errors, which are defined as innovations obtained from UMKF and the reference GAN, cumulative probabilistic tests can be designed for SL, CL and PL to evaluate if a data series corresponds to an abnormal experience

or not. Collected innovations can be used to decide when to store data for learning a new filter. Each new filter represents a new equilibrium condition which in turn encodes a set of stationary forces. Such a process is shown in Figure 3.5-c, where different types of probabilistic abnormality measurements of the switching models are estimated and thresholded. In the abnormality measurement process block, a set of abnormality tests can be considered to evaluate the detection of possible anomalies with respect to the already learned conditions of the dynamic equilibrium. This process not only ranks the innovations and computes the abnormality measurements but also facilitates the selection of the most probable model among those learned from previous experiences. The most appropriate models could be a switching Dynamical systems in case of SL and CL, while a couple of cross-modal GANs for PL. Abnormality measurements can be seen as comparable evaluations that can drive a soft decision process [84]. The former are the inputs of the abnormality detection block (see Figure 3.5-d). Such block compares anomaly signals with a threshold to detect possible deviations from previously learned models (conditions of equilibrium).

The abnormality detection procedure allows defining an incremental process similar to Dirichlet [85] (stick-breaking processes [86], Chinese restaurant [87]). Accordingly, the abnormality measurements are variables that determine the choice about whether a new experience can be described by an already available experience (learned condition of equilibrium) embedded into DBN (Figure 3.5-e) or there is a need to learn a new one (Figure 3.5-d).

Data from new experiences can be structured into multiple partitions of the state-innovation where the correlations between states and innovations facilitate clustering the new data into classes characterized by different parameters forming a new learned model. Figure 3.5-d shows the procedure of learning new models from state-innovation pairs. Such a data couple establishes a relationship between the states and an error measurement (innovations) obtained through the initial models.

Data couples can be clustered through a new learning model process. In the case of SL and CL, the new learning model process generates a set of regions which segment the state space motion and the own states depending on innovations (See Figures 3.6-d.1 and 3.6-d.2). For the PL's case, a clustering of consecutive images and their corresponding optical flows based on a similarity measurement (*i.e.*, local innovation) is considered and shown in Figure 3.6-d.3. Detected regions/clusters

Phases	Steps/Components	SL (low dimensional data)	CL (low dimensional data)	PL(high dimensional data)	Corresponding block(s)
Train (offline)	**input**	positional data	control data	first person visual data and optical-flow	b (training data series)
	initial filter	UMKF	UMKF	reference GAN (straight movement)	a
	incremental	learning motivated KFs	learning motivated KFs	detecting/learning outliers	d
	output	PGM (switching model)	PGM (switching model)	set of cross-modal GANs	c, e
	final filter/model	MJPF	MJPF	hierarchy of GANs	b (set of learned models)
Test (online)	**input**	new experience positional data series	new experience positional and control data series	new experience visual information	b (testing data series)
	output	positional state estimation	control state estimation	visual prediction	c , e
	measurement	high level innovation	high level innovation	high level innovation	c

TABLE 3.1: Phases/components of the proposed method concerning the SA modalities. (See Figure 3.6 for corresponding blocks).

can form an explicit vocabulary (in case of SDS) or implicit (in case of GANs) of switching variables such that the new learned model can address adaptively different models when it will have to evaluate new states produced by new experiences. Note that, in the case of PL, there is a need for accessing dynamic visual information to train a new set of GANs. Hence, as it shown in Figure. 3.6-d.3, for detected new clusters based on state-innovation pairs, the original data is used directly for training the new GANs. Accordingly, PL's models are all related to different effects of forces different from the one producing a straight motion in a video sequence, for example, a curving motion will generate a new GAN model.

The general framework of the proposed method described in this section includes two major phases: *(i)* an incremental off-line learning process, and *(ii)* an on-line testing procedure for detecting the possible abnormalities. Phases and components of the proposed method concerning the SA modalities are shown in Table. 3.1.

3.2.2 Mathematical modeling of SA Layers

In this section, we describe the details of the mathematical modeling for SL, CL, and PL layers of proposed SA models. Since the SL and CL have the same learning process w.r.t the low dimensional data, as shown in the Figure. 3.6 and Table. 3.1, we consider the unique formulation for both of them.

3.2.2.1 Mathematical modeling of SL and CL layers

In SL and CL, the main idea consists of learning a switching models represented by a DBN for tracking and predicting the dynamical system over time. Since both layers using low-dimensional data, we only describe the SL layer which use positional information. In case of CL, they can be changed to control system variables (i.e., steering and rotor velocity). The proposed DBN is shown in see Figure 3.7.

Dynamic modeling. As discussed in section 3.1, the SL level of SA focuses on the analysis of observed moving agents for understanding their dynamics in a given scene. SL's models $\boldsymbol{M} = \{m\}_{m=1,\ldots,M}$, employ measurements observed by the agent while performing a task. As it is well known, the dynamics of an agent can be described by hierarchical probabilistic models consisting of continuous and

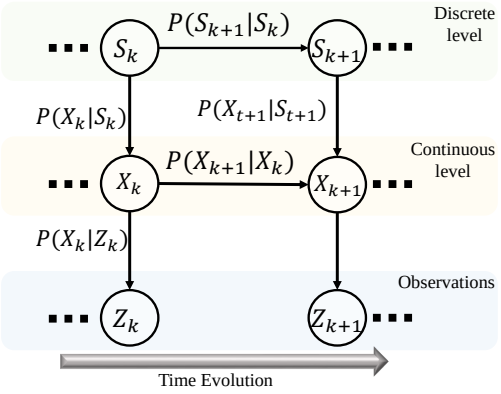


FIGURE 3.7: Proposed DBN switching models for SL and CL.

discrete random variables. Accordingly, based on equation 2.7, the dynamic model of an agent can be written as:

$$X_{k+1} = AX_k + BU_{S_k^m} + w_k, \tag{3.2}$$

As discussed in sections 2.2 and 2.2.2, X_k represents the agent's state composed of its coordinate positions and velocities at a time instant k, such that $X_k = [\boldsymbol{x}\ \dot{\boldsymbol{x}}]^\intercal$. $\boldsymbol{x} \in \mathbb{R}^d$ and $\dot{\boldsymbol{x}} \in \mathbb{R}^d$. d represents the dimensionality of the environment. $A = [A_1\ A_2]$ is a dynamic model matrix: $A_1 = [I_d\ 0_{d,d}]^\intercal$ and $A_2 = 0_{2d,d}$. I_n represents a square identity matrix of size n and $0_{l,g}$ is a $l \times g$ null matrix. w_k represents the prediction noise which is here assumed to be zero-mean Gaussian for all variables in X_k with a covariance matrix Q, such that $w_k \sim \mathcal{N}(0, Q)$.

In equation(3.2), $B = [I_2\Delta k\ I_2]^\intercal$ is a control input model and Δk is the sampling time. $U_{S_k^m} = [\dot{x}_k, \dot{y}_k]^T$ is a control vector that encodes the expected entity's velocity when its state belongs to a discrete region $S_k^m \in \boldsymbol{S}^m$. Discrete regions associated with a model m can be represented as:

$$\boldsymbol{S}^m = \{S^{m,l^m}\}_{l^m=1,\ldots,L^m}, \tag{3.3}$$

where l^m and L^m represent the index and the maximum number of superstates respectively. Additionally, a threshold value is defined where linear continuous models of superstates $\boldsymbol{S}^m$ are valid. Such a threshold is defined as:

$$\psi_{\boldsymbol{S}^m} = E(d_{\boldsymbol{S}^m}) + 3\sqrt{V(d_{\boldsymbol{S}^m})}, \tag{3.4}$$

where $d_{\boldsymbol{S}^m}$ represents a vector containing all distances between neighboring superstates, $E(\cdot)$ receives a vector of data and calculates its mean and $V(\cdot)$ its variance. The threshold value in equation(3.4) defines a certainty boundary that determines where the model is valid.

Initial model. The initial model $m = 0$ is a situation where the agent keeps the same position over time. A Kalman Filter (KF) based on an "unmotivated model"in section. 2.2 is used in tracking agents (see Figure 3.6-a.1).

The unmotivated model $m = 0$ contains only one superstate $\boldsymbol{S^0} = \{S_1^0\}$, which leads to $U_{S_1^0} \sim 0$. Hence, by relaxing $BU_{S_k^m}$ in equation 3.2, we obtain: $X_{k+1} = AX_k + w_k$, where the agent is assumed to move only under random noisy fluctuations w_k. By applying equation 2.10, innovations obtained from the initial model $m = 0$ can be collected and used for creating new models in an incremental fashion.

Creating models incrementally: As depicted in Figure 3.5-c, during the inference process, there are two different possible situations:

i) Normality: the observation can be fitted and predicted with the current learned models. In this case, there is no need to learn new models and equation 3.2 is employed for inferring future states.

ii) Abnormality: the current observation does not fit in the existing model(s), which means it is out of the boundary approximated by equation 3.4, where a random filter $U_{S_k^m} = 0_{2,1}$ is applied. In this case, it is possible to use the abnormal data for learning a new model $m + 1$. For SL, this corresponds to learning a MKF in Figure. 3.6-d.1, where the agent's dynamics can be described by quasi-constant velocity models, i.e., $U_{S_k^m} \neq 0$ in equation 3.2.

learning new switching models The state information of time instances detected as abnormal is collected and processed for learning new models. The state information can be described as: $X_k = [x_k, y_k, \dot{x}_k, \dot{y}_k]^\intercal$. Accordingly, Figure. 3.8 shows the details of learning new learning switching DBN models indicated in Figure. 3.6-d.1.

After storing the state information, the main idea is to generate a set of neurons/zones that encode similar information (quasi-constant velocities). As previously

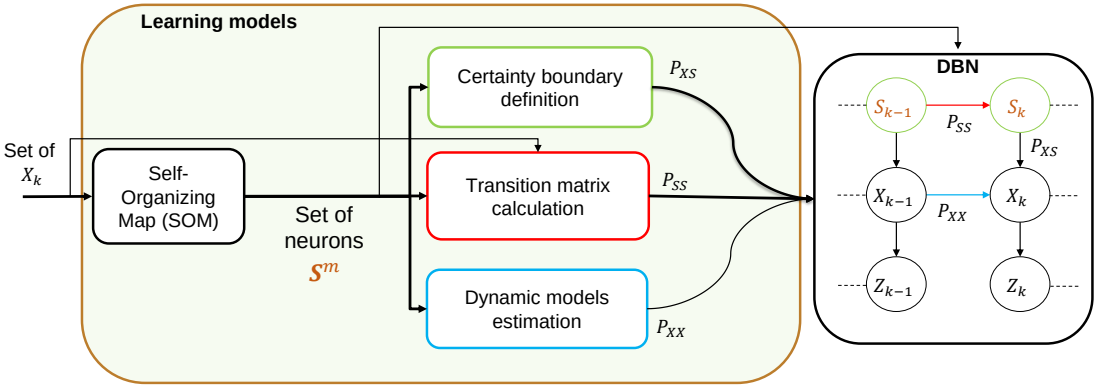


FIGURE 3.8: Generic block diagram of learning switching models.

discussed in section 3.1, GP has some limitation on modeling different dynamics appearing on the same state-space coordinate. To tackle this problem, we proposed to employ an unsupervised learning algorithm, such as Self Organizing Map (SOM) [88], which is able to differentiate and model multiple dynamics patterns appearing on the same state-space coordinate. SOM receives states X_k and generates a set of neurons such that:

$$\boldsymbol{S}^{m+1} = \{S^{m+1,l^{m+1}}\}_{l^{m+1}=1,\ldots,L^{m+1}}. \tag{3.5}$$

In the proposed SOM procedure, we use two weights, β and α, for the position (x, y) and velocity components $(\dot{x}, \dot{y})$, respectively, where $\beta+\alpha = 1$. We choose $\alpha > \beta$ to favor clustering of patterns with smaller differences in speed. Consequently, equation 3.6 shows a distance function that uses the weights employed to train the SOM, such that:

$$d(\tilde{X}, \tilde{Y}) = \sqrt{(\tilde{X} - \tilde{Y})^{\intercal} D (\tilde{X} - \tilde{Y})}, \tag{3.6}$$

where $D = [\mathcal{B}\ \mathcal{A}]$. $\mathcal{B} = [\beta I_2\ 0_{2,2}]^{\intercal}$, $\mathcal{A} = [0_{2,2}\ \alpha I_2]^{\intercal}$. $\tilde{X}$ and $\tilde{Y}$ are both 4-dimensional vectors of the form $[x\ y\ \dot{x}\ \dot{y}]^{\intercal}$.

State information X_k can be associated to the closest superstate S^{m,l^m} based on the minimum weighted distance between the measurement in question and the mean values of prototypes produced by SOM, such that:

$$\mathcal{W}_m(k) = \arg\min_m \left(d(X_k, \Psi^{m,l^m})\right), \tag{3.7}$$

where $d(\cdot,\cdot)$ is a weighted distance function between two 4-dimensional as shown in equation 3.6.

Each superstate has attached two variables: $\xi^{m,l^m} \in \boldsymbol{\xi}^m$ and $Q^{m,l^m} \in \boldsymbol{Q}^m$ related to the mean and covariance of data $\boldsymbol{X}_m$ associated to the neuron l^m by using equation 3.7. $\boldsymbol{\xi}^m = \{\xi^{m,l^m}\}_{l^m=1,\ldots,L^m}$ and $\boldsymbol{Q}^m = \{Q^{m,l^m}\}_{l^m=1,\ldots,L^m}$. Notice that ξ^{m,l^m} has the same form of $X_k \in \boldsymbol{X}_m$ since it is a result of an average process.

After obtaining the set of neurons (superstates) that represent the vocabulary of switching variables, it is possible to estimate probabilistic dependencies that are parts of the probabilistic plan model associated with the DBN described hereafter. In particular such model includes dynamic linear models, temporal probabilistic transitions matrices, and likelihood models for each element of the vocabulary (see Figure. 3.8).

Based on the dynamic model in equation 3.2, it is possible to define P_{XX} as a set of probabilistic models that capture the evolution of agents' states X_k for each SOM neuron (i.e., a vocabulary element), such that:

$$P_{XX} = \{p(X_k|X_{k-1}, S^m_{k-1})\}; \quad m = \{1, 2, \ldots, M\}, \tag{3.8}$$

where $S^m_{k-1} \in \boldsymbol{S}^m$. As mentioned before, a superstate $S^{m,l^m} \in \boldsymbol{S}^m$ is represented by the variables ξ^{m,l^m} and Q^{m,l^m} that in turn define the continuous dynamical model of agents that belong to such superstate see equation 3.2 such that:

$$U_{S_k} \sim A\xi^{m,l^m}_{(k)}; \quad Q_{S_k} \sim Q^{m,l^m}_{(k)}, \tag{3.9}$$

where $A = [0_{2,2} \;\; I_2]$.

By analyzing the activated superstates over time while executing a certain activity, it is possible to obtain a set of temporal transition matrices T^m_t. Such matrices encode the transition probabilities of passing or staying between superstates depending on the time t that the agent has spent in a superstate while model m is applied. Transition matrices facilitate the inference of next superstates given the current one, i.e., $p(S^m_k|S^m_{k-1}, t)$, such that:

$$P_{SS} = \{p(S^m_k|S^m_{k-1}, t)\}, \tag{3.10}$$

where $S^m_k \in \boldsymbol{S}^m$.

Based on equation 3.4, it is possible to define a certainty boundary where the proposed neurons, i.e., built learned models, are valid. Accordingly, if an agent's state belongs to a valid region, the dynamic model in equation 3.2 is used for predicting future states. On the other hand, when an agent's state belongs to an invalid region (empty neuron), a random filter is employed where $U_{S_k^m} = 0_{2,1}$ for estimating its future state. Based on the definition of certainty boundaries and the transition probability between regions, it is possible to define the likelihoods of agents' states to belong to a certain neuron as:

$$P_{XS} = \{p(X_k|S^m_{k|k-1})\}; \quad m = \{1, 2, \ldots, M\}, \tag{3.11}$$

where $S^m_{k|k-1}$ is a the superstate prediction given Z_{k-1}.

3.2.2.2 Mathematical modeling of PL layer

Cross-modal GAN representation. Unlike the SL and CL, the PL deals with the high-dimensional visual information observed by the agent. Namely, a sequence of images (frames) $\mathcal{I}$, and their corresponding optical-flow maps (motion) $\mathcal{O}$. To model the PL of SA, a set of cross-modal GANs [80] is trained to learn the normality from this set of visual data. In general, generative models try to maximize likelihood by minimizing the Kullback-Leibler distance between a given data distribution and the generator's distribution [89], where GANs learn it during the training process by an adversarial game between two networks: a generator (G) and a discriminator (D).

As mentioned before, a SA model not only has to be generative (capable to predict multi-level, temporal data series characterized by the previously learned knowledge), but also it needs to provide explicit measurements to evaluate or to discriminate best-fitting models of new observed sequences. In the case of GANs, Generative networks learn to perform the prediction task, wherein our case the predictions include generating the next image (frame) and optical-flow (motion map). This could be seen as a hidden state $\mathcal{X}_k^{\mathcal{P}}$ (see Figure 3.9). The update task includes having the likelihood and prediction. In GANs, the likelihood is learned and approximated by the discriminators. The outputs of the discriminators are the encoded version of optical-flow $\mathcal{D}^{\mathcal{O}}$ and image $\mathcal{D}^{\mathcal{I}}$. In our approach, Discriminators' scores are used to approximate the distance between the likelihood and the prediction.

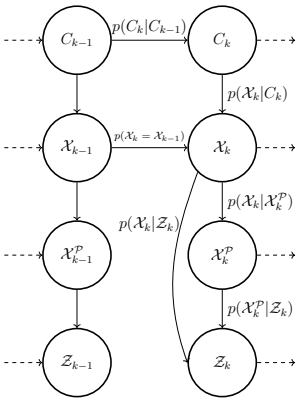


FIGURE 3.9: Proposed DBN switching models for private layer.

Intuitively, the encoded version of a given image can be seen as the state representation in the SL, while the encoded version of optical-flow represents the state derivative (motion). The error $\mathcal{E}$ can be seen as the distances between the encoded versions of prediction and observation. Following the same intuition applied in SL, we cluster the encoded version of images, motion and the error into a set of superstates. In light of the above, the states here can be defined as a function of image, motion, and the error $f([\mathcal{D}^{\mathcal{I}}, \mathcal{D}^{\mathcal{O}}, \mathcal{E}])$ for any model (superstate).

Dynamic modeling. GANs are deep networks commonly used to generate data (e.g., images) and are trained using only unsupervised data. The supervisory information in a GAN is indirectly provided by an adversarial game between two independent networks: a generator (G) and a discriminator (D). During training, G generates new data and D tries to distinguish whether its input is real (i.e., it is a training image) or it was generated by G. This competition between G and D helps to boost the ability of both G and D. For learning the conditions of equilibrium, two channels are used as observations: appearance (i.e., raw-pixels) and motion (optical-flow images) for two different cross-channel tasks. In the first task, optical-flow images are generated from the original frames. In the second task, appearance information is estimated from an optical flow image. Specifically, let $\mathcal{I}_k$ be the k-th frame of a training video and $\mathcal{O}_k$ the optical-flow obtained using $\mathcal{I}_k$ and $\mathcal{I}_{k+1}$. $\mathcal{O}_k$ is computed using [81]. For any given model $m' \in \boldsymbol{M'}$, where $\boldsymbol{M'} = \{m'\}_{m'=1,\ldots,M'}$, two networks are trained: $\mathcal{N}^{m':\mathcal{I}\rightarrow\mathcal{O}}$, which is trained to generate optical-flow from frames (task 1) and $\mathcal{N}^{m':\mathcal{O}\rightarrow\mathcal{I}}$, which generates frames from optical-flow (task 2). In both cases, inspired by [82, 83], our architecture is composed of two fully-convolutional networks: the conditional generator G and the conditional discriminator D. The G network is the U-Net architecture [82],

which is an encoder-decoder following with *skip connections* helping to preserve relevant local information.

For D, the *PatchGAN* discriminator [82] is proposed, which is based on a "small" fully-convolutional discriminator. G and D are trained using both a conditional GAN loss $\mathcal{L}_{cGAN}$ and a reconstruction loss $\mathcal{L}_{L1}$. In case of $\mathcal{N}^{m':\mathcal{I}\to\mathcal{O}}$, the training set is composed of pairs of frame-optical flow images $\mathcal{X} = \{(F_t, O_t)\}_{t=1,\dots,N}$. $\mathcal{L}_{L1}$ is given by: $\mathcal{L}_{L1}(x, y) = ||y - G(x, z)||_1$,

$$\mathcal{L}_{L1}(x, y) = ||y - G(x, z)||_1, \tag{3.12}$$

where $x = F_t$ and $y = O_t$, while the conditional adversarial loss $\mathcal{L}_{cGAN}$ is:

$$\begin{aligned}\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{(x,y)\in\mathcal{X}}[\log D(x, y)]+\\ \mathbb{E}_{x\in\{F_t\},z\in\mathcal{Z}}[\log(1 - D(x, G(x, z)))]\end{aligned} \tag{3.13}$$

In case of $\mathcal{N}^{m':\mathcal{O}\to\mathcal{I}}$, we define $\mathcal{X} = \{(O_t, F_t)\}_{t=1,\dots,N}$. Additional details about the training can be found in [82]. During the training phase of GANs, networks $\mathcal{N}^{m':\mathcal{I}\to\mathcal{O}}$ and $\mathcal{N}^{m':\mathcal{O}\to\mathcal{I}}$ learn to approximate a dynamic model for the continuous space, this can be seen as a corresponding dynamic model of SL in equation 3.2.

The discrete level uses an encoded vector $C_k^{m'} = [D^{m':\mathcal{O}\to\mathcal{I}}(\mathcal{I}_k, \mathcal{O}_k), D^{m':\mathcal{O}\to\mathcal{I}}(\mathcal{I}_k, \mathcal{O}_k)]$, where $D^{m':\mathcal{O}\to\mathcal{I}}$ and $D^{m':\mathcal{I}\to\mathcal{O}}$ are the discriminator networks of $\mathcal{N}^{m':\mathcal{O}\to\mathcal{I}}$ and $\mathcal{N}^{m':\mathcal{I}\to\mathcal{O}}$ respectively. The encoded vectors representing the expected entity's motion when its state belongs to a discrete region $C_k^{m'}$ where m' is a given model. Discrete regions of a given model m' can be represented as:

$$\boldsymbol{C^{m'}} = \{C^{m',l^{m'}}\}_{l^{m'}=1,\dots,L^{m'}}, \tag{3.14}$$

where $C_k^{m'} \in \boldsymbol{C^{m'}}$ and $L^{m'}$ is the total number of superstates for given task m'. Additionally, a threshold value is defined where linear continuous models of superstates $\boldsymbol{C}^{m'}$ are valid. The threshold can formalize as:

$$\psi_{\boldsymbol{C}^{m'}} = E(\mathcal{D}_{\boldsymbol{C}^{m'}}) + 3\sqrt{V(\mathcal{D}_{\boldsymbol{C}^{m'}})}, \tag{3.15}$$

where $\mathcal{D}_{\boldsymbol{C}^{m'}}$ contains all cross-modal discriminators likelihoods over the superstates, $E(\cdot)$ and $V(\cdot)$ are defined in equation 3.4. The threshold value in equation 3.15 is used to determine where the model is valid.

Initial model. GANs are trained in a weakly-supervised manner, the only considered supervision consists of a subset of normal data to train the first level of the hierarchy that we called *reference GANs* which corresponds to UMKF in case of SL for low-dimensional data. The *reference GANs* is trained to model a reference dynamic equilibrium where the agent being attracted by a stationary force and moves towards a fixed motivation point in a linear straight motion path. The *reference GANs* provides a reference for the next levels of the models in which all the further levels are trained in a self-supervised manner.

Similar to SL initial model, the PL initial model $m' = 0$ contains only one super-state $\boldsymbol{C^0} = \{C_1^0\}$ which contains the *reference GANs*. The detail of the training is shown in Alg. 2. The inputs of the procedure are represented by two sets: $\mathcal{Z}$ could be seen as the set of observation vectors which includes all the observations from the normal sequence of training data, and $\mathcal{V}_{m'}$ which is a subset of $\mathcal{Z}$. In case of the *reference GANs*, the initial set $\mathcal{V}_0$ is used to train two cross-modal networks $\mathcal{N}^{0:\mathcal{I}\rightarrow\mathcal{O}}$, and $\mathcal{N}^{0:\mathcal{O}\rightarrow\mathcal{I}}$. Note that, the only supervision here is for training the first model (*reference GANs*) on the initial set $\mathcal{V}_0$. The next models are built from the supervision provided by the *reference GANs*.

The mismatches between the *reference GANs* estimation and new observations lead to produce errors that in turn are used to detect new dynamic conditions of equilibrium. In other words, by using the produced errors from *reference GANs*' predictions, the model can discriminate if the dynamic equilibrium between the environment and the agent is already learned or due to a novel but again stationary forces.

Creating models incrementally. our method assumes that the distribution of the normality patterns has a high degree of diversity. In order to learn such distribution, we suggest a hierarchical strategy for high-diversity areas by encoding the different distributions into the different levels, in which, each subset of train data is used to train a different GAN. A recursive procedure is adopted to construct the proposed hierarchy of GANs. As shown in Alg. 2, the input set $\mathcal{Z}$ includes a set of coupled frame-motion maps, where $\mathcal{Z} = \{[\mathcal{I}_k, \mathcal{O}_k]\}_{k=1,\ldots,N}$, and N is the number of total train samples. Besides, the input $\mathcal{V}_{m'}$ is a subset of $\mathcal{Z}$, provided to train GANs for each model.

After training $\mathcal{N}^{0:\mathcal{I}\rightarrow\mathcal{O}}$, and $\mathcal{N}^{0:\mathcal{O}\rightarrow\mathcal{I}}$, we input $G^{0:\mathcal{I}\rightarrow\mathcal{O}}$ and $G^{0:\mathcal{O}\rightarrow\mathcal{I}}$ using each frame $\mathcal{I}$ of the entire set $\mathcal{Z}$ and its corresponding optical-flow image $\mathcal{O}$ respectively. The

Algorithm 2 Incremental training: Hierarchy of GANs

Input:

$\psi_{\boldsymbol{C}^{m'}}$: Threshold parameter for train a new GAN
$\boldsymbol{C}^{m'} = \{C^0\}$
$\mathcal{Z}$: Entire training sequences $\mathcal{Z} = \{(\mathcal{I}_k, \mathcal{O}_k)\}_{k=1,...,N}$
$\mathcal{V}_0$: Subset of $\mathcal{Z}$
$m' = 0$: Counter of models

Output:

$\{\mathcal{H}_{\boldsymbol{C}^{m'}}\}$ Hierarchy of GANs
procedure TRAINING OF CROSS-MODAL GANS
`train:`
    Train networks $\mathcal{N}^{m':\mathcal{I}\to\mathcal{O}}, \mathcal{N}^{m':\mathcal{O}\to\mathcal{I}}$, with $\mathcal{V}_{m'}$
    $\{\mathcal{H}_{\boldsymbol{C}^{m'}}\} \leftarrow$ Trained networks $\mathcal{N}^{m':\mathcal{I}\to\mathcal{O}}, \mathcal{N}^{m':\mathcal{O}\to\mathcal{I}}$
    $\mathcal{X}^{m':\mathcal{P}} \leftarrow G^{m'}(\mathcal{Z})$: predictions
    $\mathcal{D}^{m'} \leftarrow D^{m'}(\mathcal{Z})$: encoded observation
    $\mathcal{E}^{m'} \leftarrow ||D^{m'}(\mathcal{Z}) - D^{m'}(\mathcal{X}^{m':\mathcal{P}})||_1$: error
    $\mathcal{X} \leftarrow [\mathcal{D}_{m'}, \mathcal{E}_{m'}]$: states
    Clustering states: $SOM(\mathcal{X})$: superstates $\boldsymbol{C}^{m'}$
    for each identified cluster **do**
        $\mu \leftarrow$ Average score maps in this cluster
        if $\mu \geq \psi_{\boldsymbol{C}^{m'}}$ **then**
            $m' = m' + 1$
            $\mathcal{V}_{m'} \leftarrow$ Samples from cluster $C^{m'}$ in $\mathcal{Z}$
            go to `train`
    return $\{\mathcal{H}_{\boldsymbol{C}^{m'}}\}$

generators predict Frame-Motion couples as:

$$\begin{aligned} \mathcal{X}^{0:\mathcal{P}} &= \{[\mathcal{P}_k^{0:\mathcal{I}}, \mathcal{P}_k^{0:\mathcal{O}}]\}_{k=1,...,N} \\ \mathcal{P}_k^{0:\mathcal{I}} &= G^{0:\mathcal{O}\to\mathcal{I}}(\mathcal{O}_k), \quad \mathcal{P}_k^{0:\mathcal{O}} = G^{0:\mathcal{I}\to\mathcal{O}}(\mathcal{I}_k) \end{aligned} \tag{3.16}$$

where $\mathcal{P}_k^{0:\mathcal{I}}$ and $\mathcal{P}_k^{0:\mathcal{O}}$ are the k-th predicted image and predicted optical-flow respectively. The encoded versions of observations $\mathcal{Z}$ are computed by the discriminator networks D^0:

$$\begin{aligned} \mathcal{D}^{0:\mathcal{I}} &= \{D^{0:\mathcal{O}\to\mathcal{I}}(\mathcal{I}_k, \mathcal{O}_k)\}_{k=1,...,N}, \\ \mathcal{D}^{0:\mathcal{O}} &= \{D^{0:\mathcal{I}\to\mathcal{O}}(\mathcal{O}_k, \mathcal{I}_k)\}_{k=1,...,N} \end{aligned} \tag{3.17}$$

where $\mathcal{D}^{0:\mathcal{I}}$ and $\mathcal{D}^{0:\mathcal{O}}$ are the encoded version (from initial model $m' = 0$) of the observed image and observed optical-flow respectively. Similarly, the encoded distance maps $\mathcal{E}^0$ between observations $\mathcal{Z}$ and predictions $\mathcal{P}$ for both channel are

computed as:

$$\begin{aligned}\boldsymbol{\mathcal{E}}^0 &= \{[\mathcal{E}_k^{0:\mathcal{I}}, \mathcal{E}_k^{0:\mathcal{O}}]\}_{k=1,\dots,N} \\ \mathcal{E}_k^{0:\mathcal{I}} &= D^{0:\mathcal{O}\to\mathcal{I}}(\mathcal{I}_k, \mathcal{O}_k) - D^{0:\mathcal{O}\to\mathcal{I}}(\mathcal{P}^{\mathcal{I}}, \mathcal{O}_k), \\ \mathcal{E}_k^{0:\mathcal{O}} &= D^{0:\mathcal{I}\to\mathcal{O}}(\mathcal{O}_k, \mathcal{I}_k) - D^{0:\mathcal{I}\to\mathcal{O}}(\mathcal{P}_k^{\mathcal{O}}, \mathcal{I}_k)\end{aligned} \tag{3.18}$$

The distance maps $\boldsymbol{\mathcal{E}}^0$ represent a set of errors in the coupled image-motion states representation, The joint states $\{[\mathcal{D}_k^{0:\mathcal{I}}, \mathcal{D}_k^{0:\mathcal{O}}, \mathcal{E}_k]\}_{k=1,\dots,N}$ input to a self-organizing map (SOM) [88] in order to cluster similar appearance-motion information. Similar to clustering position-velocity information in the shared layer, the proposed clustering discretizes the appearance-motion representations into a set of super-states. Specifically, the SOM's output is a set of neurons encoding the state information into a set of prototypes. Detected prototypes (clusters) provide the means of discretization for representing a set of super-states, and consequently the set of superstates will be updated, such that:

$$\boldsymbol{C}^{m'+1} = \{C^{m'+1,l^{m'+1}}\}_{l^{m'+1}=1,\dots,L^{m'+1}}, \tag{3.19}$$

where $L^{m'+1}$ is the number of detected clusters (superstates) for the new model(s).

It is expected that the clusters containing the training data present a low distance score due to low innovations between predictions and observations. This is the criteria to detect the new distributions for learning new GANs in which the clusters with high average scores are considered as new distributions. The new detected distributions forming the new subsets $\mathcal{V}_l$ to train new networks $\mathcal{N}^{C^{m'}:\mathcal{I}\to\mathcal{O}}$ and $\mathcal{N}^{C^{m'}\mathcal{O}\to\mathcal{I}}$ for the new GAN models. New identified models add to the model set $\boldsymbol{M'} = \{m'\}_{m'=1,\dots,M'}$. During a task performing by the agent, all the transition matrices $T_t^{m'}$ apply in parallel.

3.2.3 Online testing: estimation and abnormality detection

After the off-line learning process, once the switching DBN models are trained, they can be used for online prediction and anomaly detection. This section describes the testing phase for state/label estimation and the proposed method for the detection of abnormalities.

3.2.3.1 Shared layer: on-line testing MJPF

Switching systems are PGMs for discrete and continuous dynamic variables in a jointly dynamical filter. Such systems have improved decision making and tracking capabilities [90]. In Switching systems, each dynamical model relating continuous state variable in successive time instants is associated with one of a discrete set of values of a random variable. The most used algorithms are the Rao-Blackwellized particle filter [91] that uses a Particle Filter (PF) in the continuous state space model and a Hidden Markov model (HMM) for the discrete space and a Markov Jump Linear Systems [92] that uses a combination of KF and PF, where the PF is used to model the discrete state space. In both cases the posterior corresponding to the joint Probability Density Function (PDF) of discrete and continuous state can be estimated at each step based on observations.

As discussed previously in section 3.2.2, the SL can be learned using a multi-level DBN switching model see Figure 3.7. Accordingly, we introduce a novel probabilistic strategy, which we call Markov Jump Particle Filter (MJPF) used for making inferences on the DBN. MJPF essentially consists of use a PF with Sequential Importance Resampling (SIR) coupled with a bank of Kalman Filters (KFs) that facilitates the inference of discrete and continuous variables jointly see Figure 3.10. The proposed MJPF consists of two inference levels.

The first level is a continuous level where states are inferred based on measurements. Predictions, $p(X_k|X_{k-1}) \in P_{XX}$, are performed by considering a bank of KFs built according to detected zones ${S^m}_k$ where quasi-constant velocity models are valid (see equation 3.2).

The second level is composed by the learned set of zones $S_k^m \in \boldsymbol{S}^m$ based on a weighted SOM (see Section. 3.2.2.1). Transitions between superstates (zones), $p(S_k^m|S_{k-1}^m) \in P_{SS}$, are used for the inference of future estimations at the discrete level by a particle filter. The relationship between both levels is done by using superstate of the particle to choose KF.

A certainty boundary differentiates among 2 cases. The first case is when a particle in the validation region might go in one of the valid neurons with probability given by an importance function $q = p(S_k^m|S_{k-1}^m)$ multiplied by the probability of being in a certain superstate after spending a duration t in S_{k-1}^m. In addition, it might go to an empty neuron with the complementary probability of the more likely

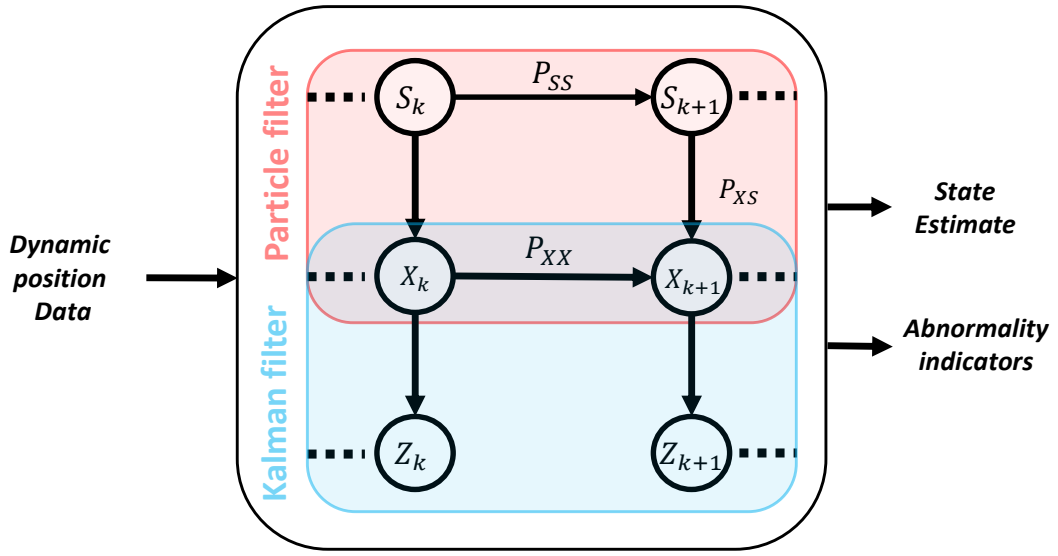


FIGURE 3.10: A Markov Jump Particle Filter (MJPF) is employed to make inference on the SL DBN.

neuron obtained using importance function. The second case is when a particle in an empty neuron (not valid region) might go in a valid region such motion can be described as the following transition probability:

$$P^m = \max(0, 1 - \frac{d(X_k, \xi^m)}{\psi_{\boldsymbol{S}^m}}). \tag{3.20}$$

For each particle, S_k^{m*}, we use a KF depending on the estimated superstate S_k^m with the control vector U (see equation 3.2) to predict $p(X_k^*|X_{k-1}^*(S^m{*}_{k-1}))$, the continuous state associated with S_k^m* and the posterior probability $p(X_k|Z_k, S_k^{m*})$ is estimated according to current measurement $Z_k \in \boldsymbol{Z}_m$, where $\mathbf{Z_k} = \{Z_1, \ldots, Z_k\}$. The update is defined as:

$$\begin{aligned} p(X_k|\mathbf{Z_k}, S_k^{m*}) &= p(X_k|Z_k, \mathbf{Z_{k-1}}, S_k^{m*}) \\ &= \frac{p(X_k|\mathbf{Z_{k-1}}, S_k^{m*})p(Z_k|, X_k^*, S_k^{m*})}{p(Z_k|\mathbf{Z_{k-1}})} \end{aligned} \tag{3.21}$$

Since Z_k and X_k are conditionally independent of $\mathbf{Z_{k-1}}$ if S_k^{m*} is known, the weight of particle S_k^{m*} is:

$$W_k^* = \frac{1}{q} \int p(Z_k, X_k^*|S_k^{m*}) dX_k^* p(S_k^*|S_{k-1}^{m*}) W_{k-1}^* \tag{3.22}$$

In SIR case of the PF in discrete space the importance function is $q = p(S_k^m|S_{k-1}^m)$. Thus, we can define

$$W_k^* = \int p(Z_k, X_k^*|S_k^{m*})dX_k^* W_{k-1}^* \tag{3.23}$$

We can write the probability inside the $\int$ in (3.23) as:

$$p(Z_k, X_k^*|S_k^{m*}) = p(Z_k|X_k^*, S_k^{m*})p(X_k^*|S_k^{m*}) \tag{3.24}$$

However, we can consider as part of the particle weight also the fact that we would like the continuous prediction within the superstate particle to be considered. To satisfy such requirement we multiply (equation 3.24) that corresponds to observation update (equation 3.21) by the prediction $p(X_k^*|X_{k-1}^*(S_{k-1}^{m*}))$

$$\begin{aligned} W_k^* = \int & p(Z_k|X_k^*, S_k^{m*})p(X_k^*|S_k^{m*}) \\ & \times p(X_k^*|X_{k-1}^*(S_{k-1}^{m*}))dX_k^* W_{k-1}^* \end{aligned} \tag{3.25}$$

By using conditional independence in (equation 3.25) we can also write

$$W_k^* = \int p(Z_k|X_k^*)p(X_k^*|S_k^{m*})p(X_k^*|X_{k-1}^*(S_{k-1}^{m*}))dX_k^* W_{k-1}^* \tag{3.26}$$

We approximate the weight as distance over the continuous state space of the two distributions (Bhattacharrya distance). A higher weight is generated with a smaller Bhattacharrya distance between prediction $p(X_k^*|X_{k-1}^*(S_k^{m*}))$ and

- probability of being inside the predicted neuron of particle $p(X_k^*|S_k^{m*})$.

$$db1 = -\ln \int \sqrt{p(X_k^*|X_{k-1}^*(S_k^{m*}))p(X_k^*|S_k^{m*})}dX_k^*; \tag{3.27}$$

- evidence $p(z_k|X_k^*)$ to have solutions near the measurement:

$$db2 = -\ln \int \sqrt{p(X_k^*|X_{k-1}^*(S_k^{m*}))p(Z_k|X_k^*)}dX_k^*; \tag{3.28}$$

where $(.)^*$ indicates the considered particle and (S_k^{m*}) means that the prediction depends on the superstate. Weights at $k-1$ can be multiplied by the inverse of

the sum of $db1$ and $db2$ for each pair of probabilities. In fact, we can approximate the weight of particle as:

$$\int p(Z_k|X_k^*)p(X_k^*|S_k^{m*})p(X_k^*|X_{k-1}^*(S_{k-1}^{m*}))dX_k^* \propto$$
$$\int \sqrt{p(Z_k|X_k^*)p(X_k^*|S_k^{m*})}p(X_k^*|X_{k-1}^*(S_{k-1}^{m*}))dX_k^* \leq$$
$$\exp^{-db1} * \exp^{-db2} = \exp^{-(db1+db2)}$$

Finally, resampling deletes particles with very low weight and clones more likely ones.

The innovation explained in section 2.2.2 in particular in equation 2.10 is considered as abnormality indicator. Subsequently, we can rewrite innovations as:

$$\epsilon_k^m = Z_k - H\hat{X}_{k|k-1}^m, \tag{3.29}$$

where ϵ_k^m is the innovation generated in the zone m where the agent is located at a time k. Z_k represents observed spatial data and $\hat{X}_{k|k-1}^m$ is the KF estimation of the agent's location at the future time instant k calculated at the time $k-1$ by using equation 3.2.

Abnormalities are moments when a tracking system (MPJF) fails to predict subsequent observations. In those cases, new models are necessary to explain new observed situations. A weighted norm of innovations is employed for detecting abnormalities, such that:

$$\mathcal{Y}_k^m = d(Z_k, H\hat{X}_{k|k-1}^m). \tag{3.30}$$

where the weighted distance $d(.)$ is defined in equation (3.6). The median abnormality indicators for all the particles is the abnormality measurement of our filter.

3.2.3.2 Private Layer: On-line testing GANs

Once the GANs hierarchy $\{\mathcal{H}_{\boldsymbol{C}^{m'}}\}$ is trained, it can be used for online prediction and anomaly detection. This section describes the testing phase for state/label estimation and the proposed method for the detection of abnormalities.

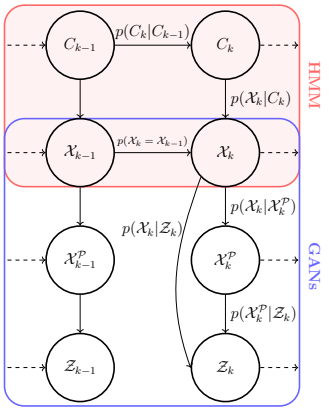


FIGURE 3.11: GANs and HMM are employed to make inference on the PL DBN

Label estimation: At testing time, we aim at estimating the state and detect possible abnormalities from the training set. More specifically, let $D^{0:\mathcal{I}\to\mathcal{O}}$ and $D^{0:\mathcal{O}\to\mathcal{I}}$ be the patch-based discriminators trained using the two channel-transformation tasks. Given a test frame $\mathcal{I}_k$ and its corresponding optical-flow image $\mathcal{O}_k$, we first produce the reconstructed $\mathcal{P}_k^{0:\mathcal{O}}$ and $\mathcal{P}_k^{0:\mathcal{I}}$ using the first level generators $G^{0:\mathcal{I}\to\mathcal{O}}$ and $G^{0:\mathcal{O}\to\mathcal{I}}$ respectively. Then, the pairs of patch-based discriminators $D^{0:\mathcal{I}\to\mathcal{O}}$ and $D^{0:\mathcal{O}\to\mathcal{I}}$ are applied for the first and the second task, respectively. This operation results in two score maps for the observation: $D^{0:\mathcal{I}\to\mathcal{O}}(\mathcal{O}_k, \mathcal{I}_k)$ and $D^{0:\mathcal{O}\to\mathcal{I}}(\mathcal{I}_k, \mathcal{O}_k)$, and two score maps for the prediction (the reconstructed data): $D^{0:\mathcal{I}\to\mathcal{O}}(\mathcal{P}_k^{0:\mathcal{O}}, \mathcal{I}_k)$ and $D^{0:\mathcal{O}\to\mathcal{I}}(\mathcal{P}_k^{0:\mathcal{I}}, \mathcal{O}_k)$. In order to estimate the state, we used equation 3.18 to generate the joint representation $\mathcal{X}_k = [\mathcal{X}_k^{\mathcal{I}}, \mathcal{X}_k^{\mathcal{O}}]$ where:

$$\begin{aligned} \mathcal{X}_k^{\mathcal{I}} &= D^{0:\mathcal{O}\to\mathcal{I}}(\mathcal{I}_k, \mathcal{O}_k) - D^{0:\mathcal{O}\to\mathcal{I}}(\mathcal{P}^{\mathcal{I}}, \mathcal{O}_k), \\ \mathcal{X}_k^{\mathcal{O}} &= D^{0:\mathcal{I}\to\mathcal{O}}(\mathcal{O}_k, \mathcal{I}_k) - D^{0:\mathcal{I}\to\mathcal{O}}(\mathcal{P}_k^{\mathcal{O}}, \mathcal{I}_k) \end{aligned} \tag{3.31}$$

For estimating the current super-state, we use $\mathcal{X}_k$ to find the closest SOM's detected prototypes. This procedure is repeated for all the models in the hierarchy $[\mathcal{H}_{\boldsymbol{C}^{m'}}]$.

This model can be seen as a switching model (see Figure. 3.11), where a hierarchy of GANs estimates the continuous space of the states whereas an HMM predicts the discrete space of the states.

Anomaly detection: Note that, a possible abnormality in the observation (e.g., an unusual object or an unusual movement) corresponds to an outlier with respect to the data distribution learned by $\mathcal{N}^{m':\mathcal{I}\to\mathcal{O}}$ and $\mathcal{N}^{m':\mathcal{O}\to\mathcal{I}}$ during training.

The presence of the anomaly, results in a low value of prediction score maps: $D^{m':\mathcal{O}\to\mathcal{I}}(\mathcal{P}^{\mathcal{I}}, \mathcal{O}_k)$ and $D^{m':\mathcal{I}\to\mathcal{O}}(\mathcal{P}^{\mathcal{O}}_k, \mathcal{I}_k)$, but a high value of observation score maps: $D^{m':\mathcal{O}\to\mathcal{I}}_l(\mathcal{I}_k, \mathcal{O}_k)$ and $D^{m':\mathcal{I}\to\mathcal{O}}(\mathcal{O}_k, \mathcal{I}_k)$. Hence, in order to decide whether an observation is normal or abnormal, we simply calculate the average value of the *innovations maps* introduced in equation 3.31 from both modalities. Therefore, the final abnormality measurement is defined as:

$$\theta_k = \overline{\mathcal{X}^{\mathcal{I}}_k} + \overline{\mathcal{X}^{\mathcal{O}}_k} \tag{3.32}$$

The final representation of PL for an observation $\mathcal{Z}_k = (\mathcal{I}_k, \mathcal{O}_k)$ consists of the computed θ_k and estimated super-state $C^{m'}_k$. We define an error threshold $\theta_{th} = \psi_{\boldsymbol{C}^{m'}}$ to detect the abnormal events: when all the levels in the hierarchy of GANs classify the sample as abnormal (e.g., dummy super-state) and the measurement θ_k is higher than this threshold, the current measurement is considered as an abnormality. Note that the process is aligned to the one followed with SL layer with the advantage that GANs deal with high multidimensional inputs as well as with not linear dynamic models at the continuous level. This complexity is required to analyze video data involved in PL.

3.2.4 Experimental results

Dataset and evaluation scenario: In order to test the proposed method for abnormality detection, it is considered the real dataset acquired from real vehicle that described in subsection 2.6.1.1. In this section we introduce another scenario called U-turn where the vehicle performs a perimeter monitoring and faces a pedestrian, so it makes a U-turn to continue the task in the opposite direction as shown in Figure 3.12.

In addition to the positional and FPV information used previously, in this section, we employ internal information of the vehicle to test CL of SA. Such information is related to the controls of the vehicle's motion, i.e., steering angle s_k and rotors' velocity v_k. Accordingly, the state information of the control modality can be described as: $X_k = [s_k, v_k, \dot{s}_k, \dot{v}_k]^\intercal$.

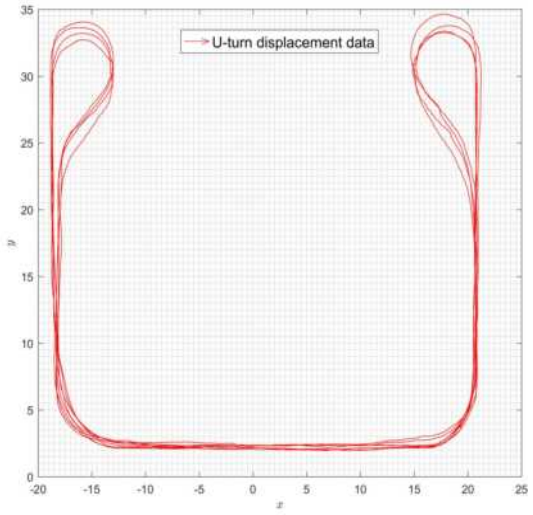


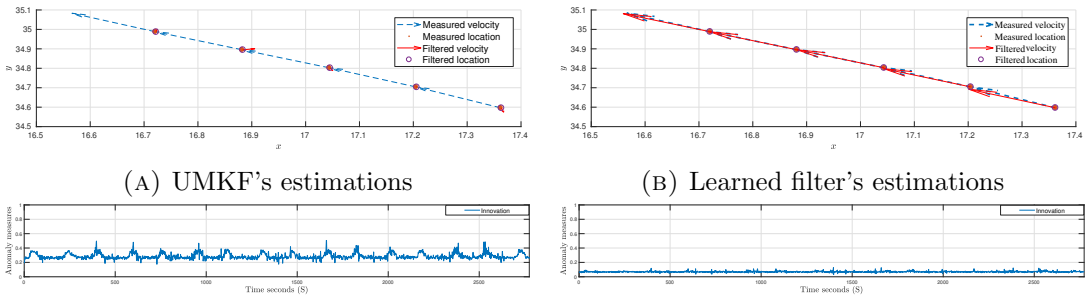
FIGURE 3.12: Displacement data for U-turn scenario used for testing abnormalities in vehicle behavior

3.2.4.1 Training SA layers

The dynamics of the scenario I in subsection 2.6.1.1 are used as a training set to learn models (SL, CL, and PL). For each layer, we use different observations. Accordingly, SL uses positional information, and CL employs control information while PL utilizes first-person visual data. SL and CL are modeled by MJPF, whereas PL is modeled as a hierarchy of GANs.

Initial model in SL. As we discussed in Section. 3.2.1, the reference filter for the shared layer is a UMKF, see Figure 3.6-(a.1), which assumes the simple condition of equilibrium in which the agent is not moving. A sequence of state estimation samples of UMKF is shown in Figure 3.13a. In this figure, UMKF always predicts the agent will stay in a still position, including a negligible perturbation error, see small velocity vectors in red. However, this assumption is not true in the observed data, (see large velocity vectors in blue in Figure 3.13a, and leads to large innovation values such as shown in Figure 3.13c).

Training incremental models in SL. By applying the reference filter UMKF over the training data a set of innovation values can be obtained. This set of innovation values plotted in Figure 3.13c, and is used in the next training iteration to incrementally learn a set of new filters, as described in Section. 3.2.2.1 (see Figure 3.6-(d.1)). Such a set of learned filters (MKFs) encode models that describe new conditions of equilibrium. Estimations from learned filters perform more accurate estimations concerning the simple UMKF when dealing again with similar



(A) UMKF's estimations (B) Learned filter's estimations

(C) UMKF's innovations (D) Learned filter's innovations

FIGURE 3.13: SL state estimation: (a) and (b) correspond to the estimations made by the UMKF and the learned filter respectively. Abnormality signals in SL: (c) and (d) show the innovations generated by the UMKF and the learned filter respectively.

abnormal situations, compare blue and red arrows in 3.13b. It can be seen that predictions are close to the observations, producing low errors, i.e., low innovation values, as shown in Figure 3.13d.

Initial model in PL: As mentioned in Section. 3.2.2.2, constructing the GAN hierarchical model is done based on the distance of discriminators scores between the predictions and the real observations. The first level of GANs (*reference GANs*) is trained on a selected subset of normal samples from *perimeter monitoring* sequence. This subset represents the captured sequences while the vehicle moves on a straight path in a normal situation, i.e., when the road is empty, and the vehicle moves straight. The hypothesis is that this subset only represents one of the motion distributions and appearance in a highly diverse data condition. As a result, when the pair of *reference GANs* detects an abnormality in the corresponding set on which is trained, the corresponding observations can be considered as outliers. This hypothesis is confirmed by testing the *reference GANs* over the entire sequences of perimeter monitoring and observing the discriminators' scores distances between the prediction and the observation. Figure 3.14 shows the results of training *reference GANs*. Our hypothesis concerning the complexity of distributions is confirmed in Figure 3.14 (a) where the test is performed using only the *reference GANs*.

Training incremental models in PL: As shown in Figure 3.14, *reference GANs* can predict/detect the straight path (white background area) perfectly. On the other hand, when the vehicle curves (green bars), it fails and recognizes curving as an abnormal event. This means the *reference GAN* discriminators' scores distances between the prediction and the observation (abnormality signal) are higher

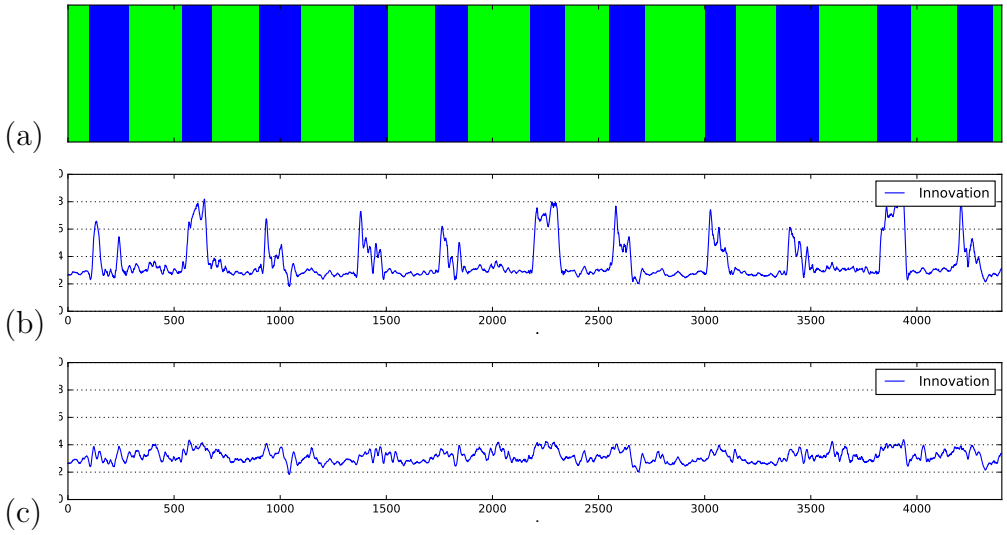


FIGURE 3.14: Training hierarchy of GANs: (a) ground truth labels, the green background means the vehicle moves on a straight path, while the blue bars indicate curving. (b) and (c) show the signal of the averaged score distance values between prediction and observation (innovation) for the first level of GANs and the hierarchy of GANs, respectively. The horizontal axis represents time and the vertical axis is the innovation values.

over the curving areas which was expected. However, after collecting this set of abnormal data and training the second level of GANs, the new learned models facilitate to recognize the entire training sequence as normal. The estimation of optical-flow and frame for each level (iteration) of the training process is shown in Figure 3.15. For each case an image triplet is shown, where the left image is the

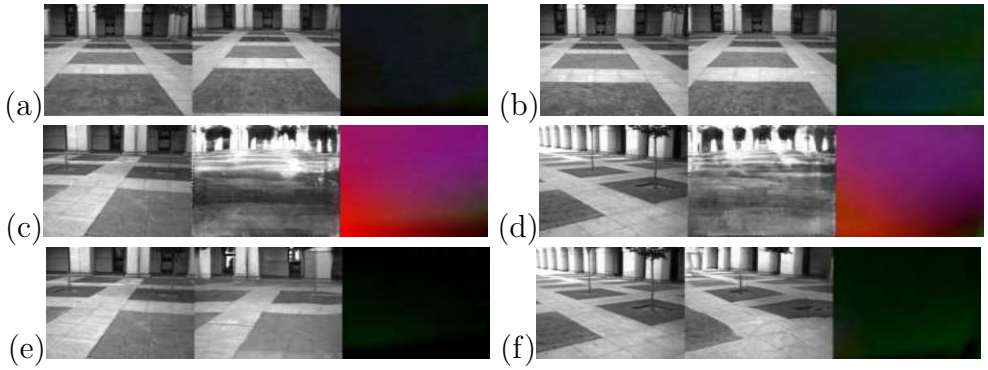


FIGURE 3.15: PL state estimation: (a) and (b) are the estimated video frame and optical-flow motion map from the *reference GAN* while the vehicle moves toward a straight path. (c) and (d) are the estimations from the *reference GAN* while the vehicle curving, (e) and (f) are the same samples predicted by the hierarchy of GANs after training. Each triplet image contains the ground truth observation (left), the predicted frame (center), and the difference between prediction optical-flow map and the ground truth where black pixels indicate a high accuracy at the prediction stage.

ground truth observed frame, the central image shows the predicted frame, and the right image is the difference between the observed optical-flow motion map and the predicted optical-flow. The lower is the distance between predicted motion and observation the blacker (values are near to "0") the right image. In this figure, (a), (b), (c), and (d) show the output estimation for the initial GANs. As can be seen, straight motions displayed in (a) and (b) are correctly estimated, (see the low error, i.e., black pixels, in the right frames of their triplets). Nonetheless, the initial model is unable to predict curve motions shown in (c) and (d), (see the high error, i.e., colorful pixels, in the right frames of their triplets). Figure 3.15 (e) and (f) show the estimation from the hierarchy of GANs after a full training phase in case of curving. It can be observed this time how the GANs estimate the curving motion with high accuracy, (see the number of black pixels in the triplet's right frames).

As the *reference GANs* are trained on the reference situation which is the straight movements (see Figure 3.6-(a.3)), therefore is it expected to have a good estimation while the vehicle moving straight (Figure 3.15-a-b). However, this filter fails to estimate curves (Figure 3.15-c and d). The incremental nature of the proposed method is demonstrated in Figure 3.6-(d.3), where low estimation errors are obtained by using the second level GANs for predicting curve motions, (see 3.15-e-f).

3.2.4.2 Final learned filter for normality representation

After training SA layers over the training sequence, to evaluate the final learned models, we select a period of the normal perimeter monitoring task (see Figure 3.16-a) as a test scenario. As reviewed in section 3.2.1 both SL and PL, represent their situation awareness by a set of superstates following and abnormality signals.

(A) perimeter monitoring (B) U-turn

FIGURE 3.16: Sub-sequence examples from testing scenarios

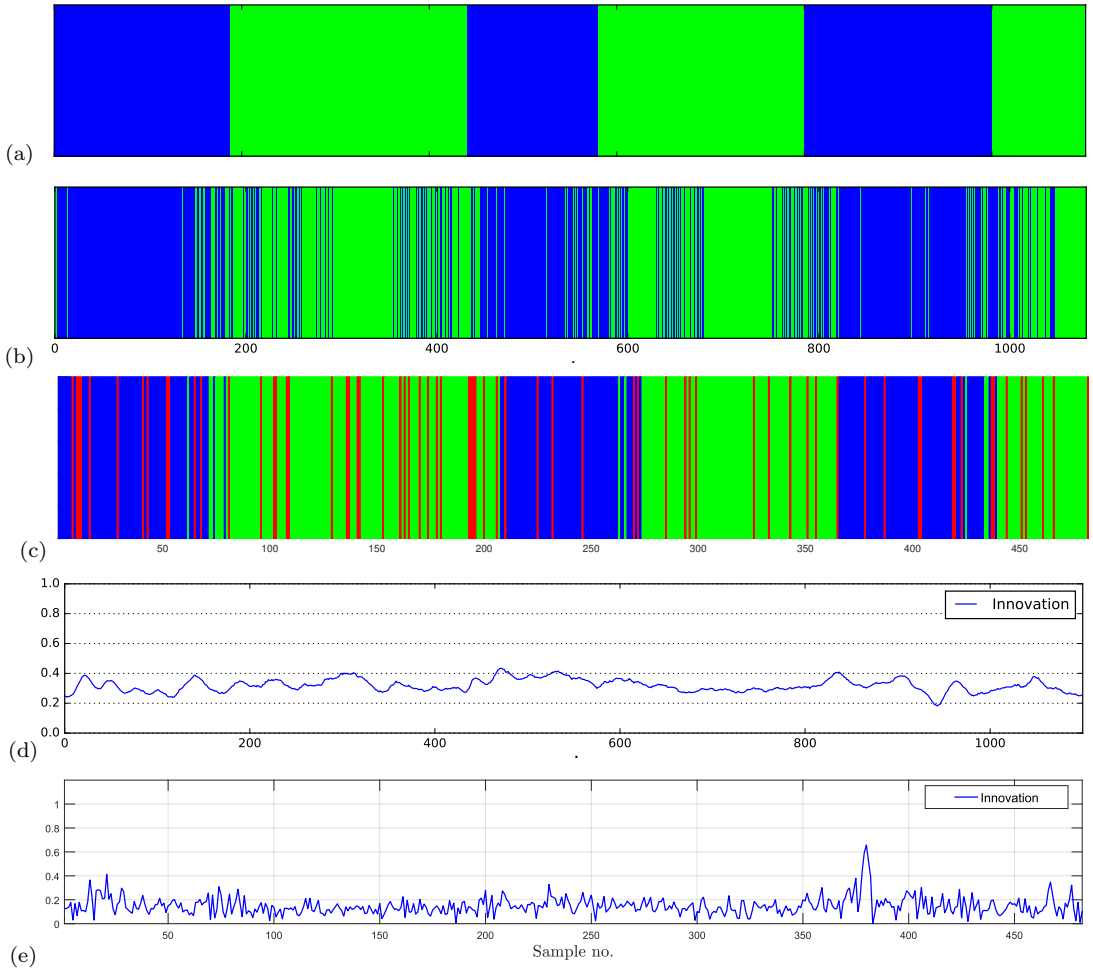


FIGURE 3.17: Normality representations of PL and SL: (a) shows the ground truth labels, moving straight is green and blue bars represent curving. Color-coded super-states sequences $\{C_k^{m'}\}$ and $\{S_k^m\}$ are shown in (b) and (c) respectively. They are highly correlated with the agent's real status (a). Images (d) and (e) show the abnormality signals from PL and SL, respectively. The horizontal axes in (d) and (e) represent the sample number, and the vertical axes show the abnormality signals.

This set of results for PL and SL is shown in Figure 3.17 which simply visualizes the learned normality representations. The ground truth label is shown in Figure 3.17-a, and the color-coded detected super states from PL $\{C^{m'}\}$ and SL $\{S^m\}$ are illustrated in Figure 3.17-b and Figure 3.17-c respectively. It clearly shows that the pattern of superstates is repetitive and highly-correlated with the ground truth. It also shows a strong correlation between the sequence of PL and SL superstates.

The study of the cross-correlation between the SL and PL is beyond the scope of this work, but it is also interesting to demonstrate such relation. For showing the correlation of two learned models (SL and PL), we divided the environment

into eight meaningful zones including curves and straight path. This semantical partitioning of state-space is shown in Figure 3.18. For the training scenario (normal situation) the color-coded super-states of SL and PL are visualized over the environment plane.

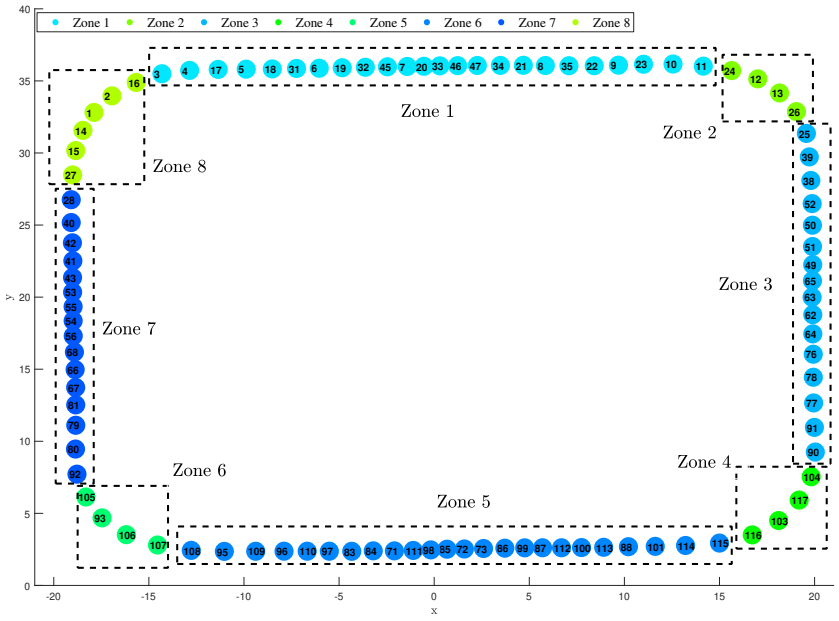


FIGURE 3.18: Color-coded zones from SL and PL.

3.2.4.3 Abnormality detection in dynamic data series

We performed an online testing setup to evaluate the performance of our models. Accordingly, we select a period of the U-turn task as test scenario (see Figure 3.16-b). The goal is to detect the abnormality consisting of the presence of the pedestrian and consequently the unexpected action of the agent with respect to the learned normality during the perimeter monitoring. Figure 3.19 shows the result of anomaly detection from PL, SL and CL. In Figure 3.19-a, the green background represents a vehicle's straight path, the blue bars indicate curving, and red bars show the presence of an abnormal situation (which corresponds to the static pedestrian). The abnormality area starts at first sight of the pedestrian, and it continues until the avoiding maneuver finishes (end of U-turn).

The abnormality signal generated by PL, as shown Figure 3.19-b, is computed by averaging over the distance maps between prediction and observation score maps: when an abnormality arises, the proposed measure does not undergo large changes since a local abnormality (see Figure 3.20-c) can not change the average

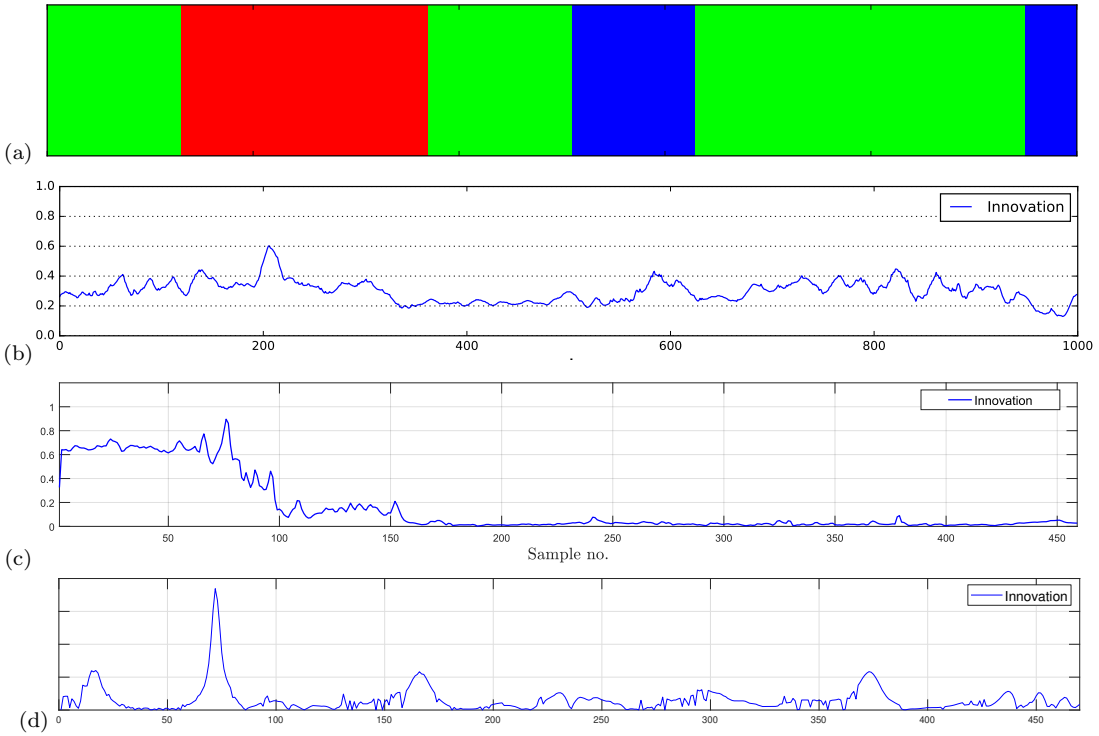


FIGURE 3.19: Abnormality in the U-turn scenario: (a) ground truth labels. (b), (c) and (d) generated abnormality signal (innovation) from PL, SL and CL, respectively. The horizontal axis represents the sample number, and the vertical axis shows the innovation values (abnormality signal).

value significantly. However, as soon as it is observed a full sight of the pedestrian and the vehicle starts performing the avoidance maneuver, the abnormality signal becomes higher since both observed appearance and action represent unknown situations. This situation is shown in Figure 3.20-(d,e). As soon as the agent back to the known situation (e.g., curving) the abnormality signal becomes lower. The abnormality signal (innovations) generated by SL is shown in Figure 3.19-c. The abnormality produced by the vehicle is higher due to the opposite velocity compared with the normal behavior of the model. In the control module, high peaks in the abnormality signal (see Figure. 3.19-d) are due to the presence of unseen maneuvers in the steering and rotors with respect to the perimeter monitoring task. The CL does not detect the motion in the opposite direction as abnormality because steering and rotors of the vehicle present similar dynamics to the ones already experienced in the perimeter monitoring task. Therefore, we get high peaks only in the u-turn maneuver and curves which show high deviation from the training model.

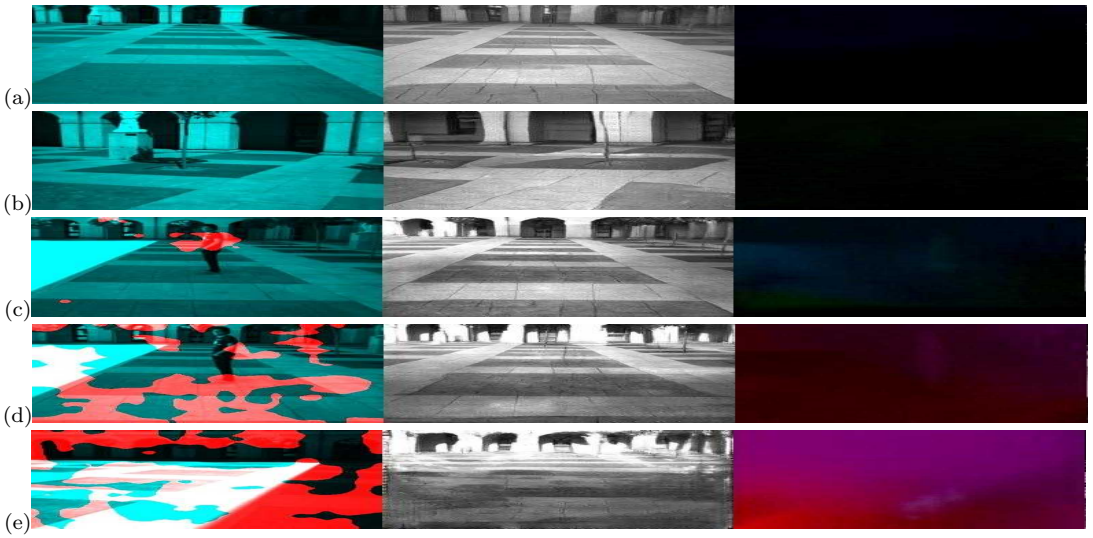
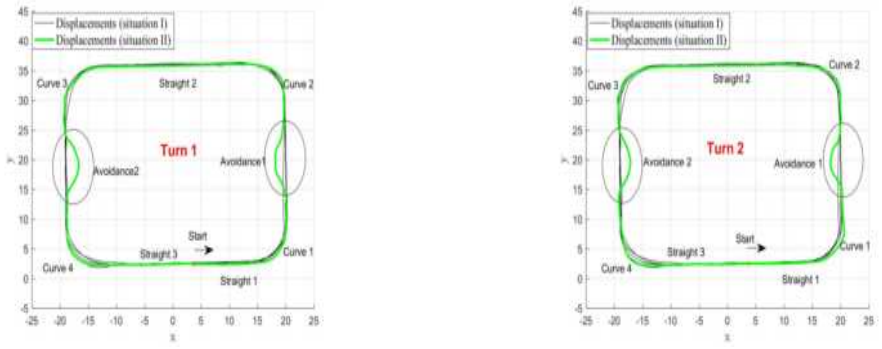


FIGURE 3.20: Visualization of abnormality: the first column shows the localization over the original frame, the second column is the predicted frame, and the last column shows the pixel-by-pixel distance over the optical-flow maps. (a) moving straight, (b) curving, (c) first observation of the pedestrian, (d) and (e) performing the avoiding action.

Additional abnormality measurements: Abnormalities are deviations from learned behaviors in a given environment. An abnormality shows itself when multilevel predictions are not confirmed by new incoming observations. In the multilevel filter described in section 3.2.3.1 this can happen at continuous and discrete levels, so that multiple abnormality indicators can be defined. Accordingly, $db1$ and $db2$ in equations (3.27) and 3.27 respectively are also considered as indicators for abnormality detection. The value of $db1$ relates to the similarity between prediction of the state and the likelihood to be in the predicted superstate, i.e. indicates if particles are coherent with the semantic discrete prediction of the learned plan. The value of $db2$ relates to the similarity between the state prediction and the continuous state evidence related to the new observation in each superstate.

Abnormality detection examples in SL by using innovation, $db1$ **and** $db2$ **indicators**: We identify abnormalities by observing new data that do not correspond to learned perimeter monitoring model. To detect the abnormality three thresholds are considered: $db1 = 0.2$, $db2 = 0.3$ and $\mathcal{Y}_k = 0.3$. The values for the thresholds are obtained by computing mean and variance for each parameter $db1$, $db2$, and $\mathcal{Y}_k$ using the training set (perimeter monitoring) for testing the method. Scenarios discussed previously (i.e., Pedestrian avoidance, and U-turn) include unseen maneuvers. Figures 3.21a and 3.21b show the observations of



(A) First turn (B) Second turn

FIGURE 3.21: Observation data related to pedestrian avoidance.

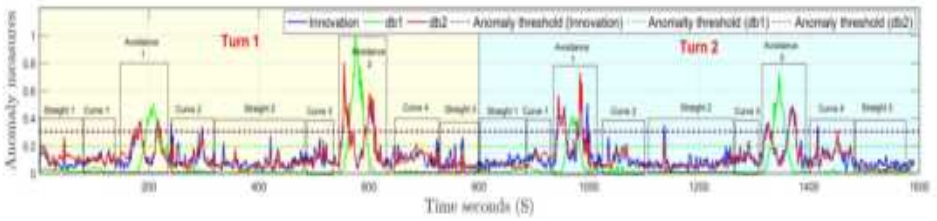
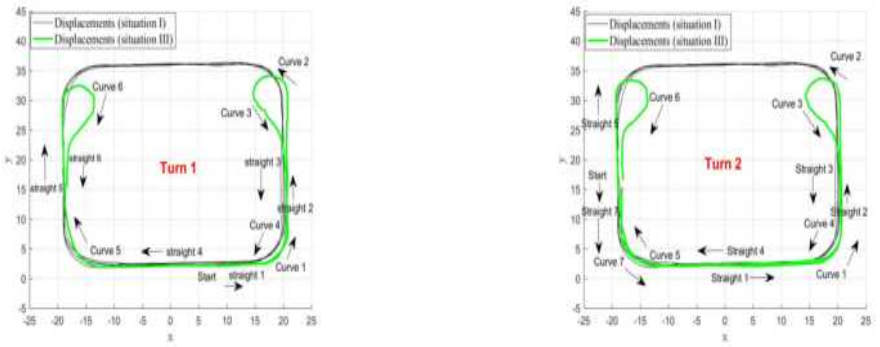


FIGURE 3.22: Abnormality measurements through time for perimeter control activity with avoidance of static pedestrians.

the pedestrian avoidance maneuver with respect to the states of the task used for training. High indicators suggest the presence of unseen dynamics (i.e. anomalies). Figure 3.22 characterizes the abnormality corresponding to pedestrian avoidance during perimeter monitoring. With aforementioned abnormality thresholds, it is possible to find time shots in which peaks in $db1$ indicate the current experience is outside the learned model. Innovation and $db2$ anomaly indicators are high in such time shot due to the difference between expected prediction (for which going straight has higher probability) and the likelihood behavior in a curved path. As one can see in Figures 3.21a and 3.21b on both turns, the likelihood follows well the predicted motion in zones different from the avoidance regions and no abnormalities are present in such time intervals in Turn 1 and 2 (Figure 3.22).

In order to analyze the U-turn maneuver experiment, we considered two turns as shown in Figure 3.23a and 3.23b, where the observations of such experiment are plotted with respect to the states of the trained task. Figure 3.24 represents Turn 1 corresponding to Figure 3.23a: the abnormality produced by the vehicle starting to turn back becomes evident from the zone of Curve 2. A higher peak of $db1$ is present in Curve 3 due to the fact that the observations are outside the domain of superstates trained for perimeter monitoring. Figure 3.23a shows the reason for which KF innovation and $db2$ become higher in the same time interval



(A) First turn (B) Second turn

FIGURE 3.23: Observation data related to U-turn

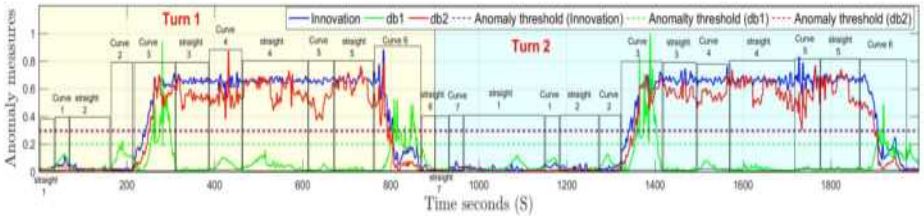


FIGURE 3.24: Abnormality measurements through time for perimeter control activity with U-turn.

due to the opposite velocity compared with the normal behavior of the model. Figure 3.24 shows that innovation and $db2$ remain high until the zone of Curve 6 because the direction changes in the U-turn case. Instead $db1$ from zone Straight 3 until zone Straight 5 is low. This is because the distance between prediction and probability of the superstate is low as we fall in a region crossed during perimeter monitoring reference experience: the predicted states between zone Straight 3 until zone Straight 5 are similar to those learned. Again $db1$ becomes higher in the zone of curve 6 as presented in Figure 3.23a. As one can see, the normality is present from zone of straight 6.

3.2.5 Discussions

In order to improve the single modality awareness models, this chapter has presented a multi-perspective approach to detect anomalies for moving agents. The proposed models consider two levels (shared and private) that handle different types of information from a dynamic agent. SL uses a state-space representation from an external observer whereas CL and PL employ a state-space representation from the analyzed agent. We proposed two approaches for modeling our multi-perspective awareness:

- Semi-supervised GP-based: Generates locally uniform motion models by dividing a Gaussian process that approximates agents' displacements on the scene and provides a SL SA based on EC models. Such models are then used to train in a semi-unsupervised way a set of GANs that produce an estimation of external and internal parameters of moving agents. The use of a GP approach together with a superpixel algorithm are not able to distinguish different dynamic in the same spatial location. Furthermore, the generation of zones needs a combinations of algorithms which increases the complexity.

- Unsupervised incremental learning: In order to solve the issues with GP-based approach, an unsupervised clustering technique is proposed as SOM to generate and detect meaningful zones. Detected zones are then used for train a set of GANs. A multi-level switching dynamic model is proposed to learn the SA layers from different sensorial data. Learning SA models is based on incremental process, where the properties of such modeling are summarized in Table 3.2.

Furthermore, based on our observations of the current model, it is clear that the SL, CL and PL levels are providing complementary information regarding the situation awareness. As an instance, it has been observed that PL's superstates are invariant to the agent's location, while SL's superstates representation is sensitive to such spatial information. In other words, PL representation can be seen as the semantic feature of the agent's states awareness of itself (e.g., moving straight, curving) regardless of its current location. In contrast, the SL representation includes spatial information of external surrounding states, which generates more specific superstates for describing each zone, see Figure 3.18. For generating artificial aware agents, it is proposed to embed the sense of SA (understanding of own states) and situation awareness (comprehension of external surrounding states) in the entity in question.

In light of the above, combining information from different sources (SA layers) for decision making can robust the SA model. In particular, situational awareness and self-reactions are complementary information, where they can be used to model the causality between different layer as interaction cross-correlations. A coupled Bayesian network could represent such interaction model, enabling a potential improvement on the detection of abnormality and consequently boost the entire SA model.

In the next chapter, we focus on modeling causality as an interaction integration of different sensorial modalities, and we study the usefulness of such modeling. The ultimate goal is to develop intelligent awareness systems that can concisely summarize their beliefs about the world with diverse predictions, integrate information and beliefs across different components of awareness to extract a holistic view of the world, and explain why they believe what they believe.

Self-awareness model properties	**Description**
Generative modeling	Starting with an initial model, new models incrementally created as new experiences. The derived models are further able to generate future state predictions at different abstraction levels using probabilistic inference techniques such as MJPF, GANs.
Discriminative modeling	By detecting and using abnormalities, our models can identify the fittest model wrt. current observations, use it for predictions and eventually create a new model that encodes the detected abnormalities.
Hierarchical modeling	The proposed DBNs are composed of at least two levels of inference: *i)* a continuous inference based on state information obtained from observations and *ii)* a discrete inference base on discrete variables that encode certain dynamics in state regions for multi-modal models. The continuous variables depend on discrete ones, which facilitates a hierarchical Bayesian representation.
Temporal reasoning	Inferences of the future based on the current contextual information. Additionally, the proposed DBN reasoning implies a description of temporal causalities between different states at different inference levels.
Uncertain reasoning	The selected Bayesian representation, facilitates the inferences of random variables at different inference levels. Such representation of uncertainties enables to define abnormalities in a general probabilistic.

TABLE 3.2: Properties of the proposed Self-awareness model.

Chapter 4

A Unified Interaction Multi-Modal Awareness System

This chapter proposes a methodology for representing and modelling interactions among multisensory data for prediction purposes. We describe how a coupled representation can learn appropriate models. Additionally, we show how the MJPF can be adequately employed to predict its internal and environmental states as well as to distinguish among normal and abnormal behaviours. Learning and testing phases are discussed for two different protocols. Measurements for comparing the performances of predicting algorithms are also included and described in this chapter.

In particular, through this chapter we propose two different representation of Coupled Dynamic Bayesian Networks (C-DBN) for modeling the interaction between modalities. Furthermore, in order to validate the above proposed methodology, we consider the real dataset as mentioned in section 2.6.1.1 to model the causality between different sensorial information (i.e., between proprioceptive and exteroceptive signals acquired by the AA). Finally, we discuss and present a multi-agent interaction scenario over a simulated dataset to test the proposed method for modelling causality of moving agents.

4.1 Generation of states

Let Z_k be the observations of multisensorial data at a time instant k such that:

$$Z_k = \left[\mathcal{Z}_k^d\right]_{d=1,\dots,D} \tag{4.1}$$

where $\mathcal{Z}_k^d$ represents a one-dimensional measurement indexed as d at the time instant k. D is the total number of dimensions of the multi-sensory data.

Let X_k be the states of multisensorial data at a time instant k such that:

$$X_k = \left[\mathcal{X}_k^d\right]_{d=1,\dots,D} \tag{4.2}$$

where $\mathcal{X}_k^d$ represents the state of the one-dimensional measurement $\mathcal{Z}_k^d$, such that:

$$\mathcal{Z}_k^d = \mathcal{X}_k^d + \nu_k^d, \tag{4.3}$$

where ν_k^d encodes the observation noise introduced by the sensor from which $\mathcal{Z}_k^d$ is obtained.

This work considers Generalized States (GSs) [93] that carry information of the first-time derivative of traditional states. Such first-time derivatives are approximated based on the differences between consecutive observations such that $\dot{\mathcal{X}}_k^d \sim \frac{\mathcal{Z}_{k+1}^d - \mathcal{Z}_k^d}{\Delta k}$. Hence, a vector consisting of multisensorial data derivatives can be written as follows:

$$\dot{X}_k = \left[\dot{\mathcal{X}}_k^d\right]_{d=1,\dots,D-1}. \tag{4.4}$$

At each time instant, GSs are defined as the concatenation of the agent's states (see Equation 4.2) and their time derivative (see Equation 4.4) such that:

$$\tilde{X}_k = \begin{bmatrix} X_k \\ \dot{X}_k \end{bmatrix} = \begin{bmatrix} \mathcal{X}_k^d \\ \dot{\mathcal{X}}_k^d \end{bmatrix}_{d=1,\dots,D-1}. \tag{4.5}$$

In other words, at each time instant, the proposed methodology models multi-sensory observations Z_k as GSs $\tilde{X}_k$ consisting of a $2 \times d$-dimensional array where d defines the number of dimensions for multi-sensory observations.

4.2 Generation of modalities

Multi-sensory data is divided into modalities depending on the scope and statistical properties of the measured information. As recent works suggest [94–96], analyzing multimodal sensory data can be beneficial for modeling more realistic scenarios where a variety of information is available through time. The combination of multimodal sensor data and the modeling of interactions between them have proved to improve the accuracy of several processes such as inferences, detection of anomalies and task-classification. Motivated by the benefits of modeling multimodal data, This work considers multi-sensory modalities such that:

$$Z_k^m = \left[\mathcal{Z}_k^{d_m}\right]_{d_m=1,\dots,D_m}, \tag{4.6}$$

where $m = 1, 2, \dots, M$ indexes the identified modalities and M is the total number of them. d_m is a variable that encodes all dimensional measurements d associated with the same modality m. In other words, equation 4.6 groups at each time instant the measurements in equation 4.1 belonging to the same modalities. Similarly, GSs in equation 4.5 can be rewritten as:

$$\tilde{X}_k^m = \begin{bmatrix} X_k^m \\ \dot{X}_k^m \end{bmatrix} = \begin{bmatrix} \mathcal{X}_k^{d_m} \\ \dot{\mathcal{X}}_k^{d_m} \end{bmatrix}_{d_m=1,\dots,D_m}, \tag{4.7}$$

where X_k^m and $\dot{X}_k^m$ are respectively the states and their first-time derivatives related to the modality m.

4.3 Learning phase: Probabilistic models for multisensory data

Let $\boldsymbol{Z}_{train}$ be a set of consecutive observations used for learning predicting models such that $\boldsymbol{Z}_{train} = \left\{Z_k\right\}_{k=1,\dots,K}$. By grouping $\boldsymbol{Z}_{train}$ into modules, it is possible to write the training data as $\boldsymbol{Z}_{train}^m = \left\{Z_k^m\right\}_{k=1,\dots,K}$ and its correspondent GSs as $\tilde{\boldsymbol{X}}_{train}^m = \left\{\tilde{X}_k^m\right\}_{k=1,\dots,K}$ (refer to equations 4.6 and 4.7).

This work comes to code GSs into discrete random variables that can be employed to predict future time instances. In Chapter 3, we used SOM to cluster dynamic

data and generate models for state estimation purposes. Nonetheless, a disadvantage of SOM lies on the generation of nodes where the membership probability of training data, e.g. dead nodes that do not participate in the system's inferences but utilize its resources. To overcome this problem, Growing Neural Gas (GNG) with the utility measurement [97] is proposed in this work.

As mentioned previously, two main protocols for coding and predicting multisensory data are proposed and evaluated:

- *Separate Approach (S-A)*
- *Joint Approach (J-A)*

The codification approach for each of them is explained in the coming two sections.

4.3.1 Separate approach (S-A)

This protocol consists of a two-step codification scheme for grouping multimodal GSs (see Figure 4.1). Initially, this protocol performs a separate codification of GSs belonging to different modalities. Afterwards, it performs a further codification where generated clusters are combined based on their frequency of simultaneous activation.

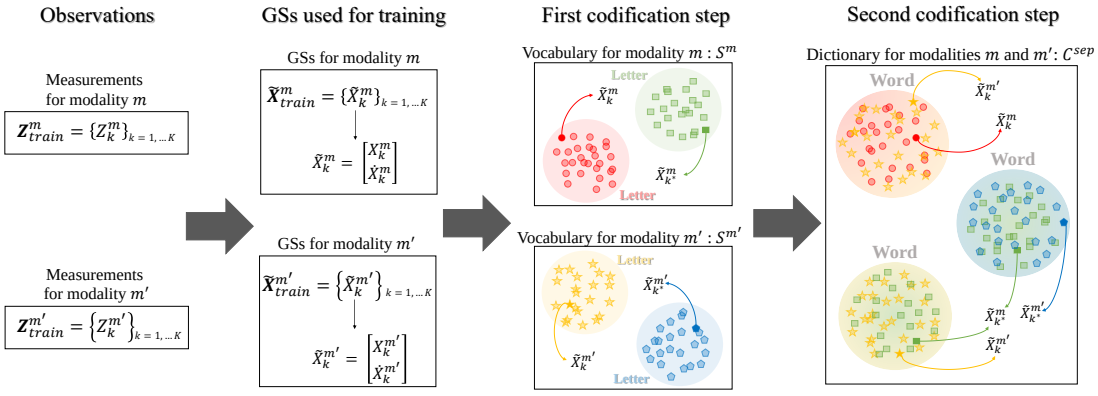


FIGURE 4.1: Two-step codification process for generating discrete states in the S-A clustering protocol. Observations from bi-modal (m and m') information produce GSs that are independently grouped into clusters (letters) for each modality. Subsequently, obtained clusters are joint, generating a set of words that capture bi-modal behaviors into discrete components. Note that k and k^* represent different time instants belonging to the training set.

***First codification**:* GSs data designed for training, $\tilde{\boldsymbol{X}}^m_{train}$, is clustered independently through an unsupervised clustering (GNG) algorithm for each modality. This process implies a total of M different clustering processes that in turn generate a total of N_m clusters each where m indexes the m^{th} modality. A cluster related to the modality m can be written as S^{n_m} where n_m indexes the n^{th} cluster of the modality m. Accordingly, each cluster is seen as a *letter*, it is possible to define the *vocabulary* associated with the modality m as:

$$\mathcal{S}^m = \{S^{n_m}\}_{n_m=1,\dots,N_m}. \tag{4.8}$$

Note that $\mathcal{S}^m$ defines an independent semantics for each modality. Such semantics encodes state-space regions where dynamical models defined by clustered time-derivative information are valid.

***Second codification**:* Since the co-occurrence and conditional dependencies between modalities are relevant subjects of study in this work, the simultaneous activation of clusters (letters) from different modules is studied and codified. For that purpose, let $S^m_k \in \mathcal{S}^m$ be the activated cluster of the modality m at the time instant k in the training set. Note that such a cluster is related to the GS $\tilde{X}^m_k \in \tilde{\boldsymbol{X}}^m_{train}$ and observation $Z^m_k \in \boldsymbol{Z}_{train}$. The simultaneous activation of vocabulary letters from different modalities can be seen as a *word* defined as $C^{sep,l} = \{S^{n_m}\}_{m=1,\dots,M}$ where l indexes the word in question. Thus, it is possible to define a *dictionary* $\mathcal{C}^{sep}$ as follows:

$$\mathcal{C}^{sep} = \{C^{sep,l}\}_{l=1,\dots,L}, \tag{4.9}$$

$\mathcal{C}^{sep}$ defines a dependent semantics where modalities' states are represented into single variables (words) where L represents the total number of observed words formed by combining multimodal vocabularies. Such *dictionary* represent the interaction among modalities. Since L is generally large, we consider a simple thresholding process that selects the most frequent simultaneous activation.

To do such a process, let f^l be the frequency of the word C^l based on the training data. Additionally, let l^* be subspace of words l where $f^l > f_{th}$. The variable f_{th} is a threshold that limits the minimum frequency that a word must have to belong to l^*. Such a threshold is fixed as $f_{th} = \mu(F) - 3\sigma(F)$, where $F = \left\{f^l\right\}_{l=1,\dots,L}$, $\mu(\cdot)$ and $\sigma(\cdot)$ are functions that extract respectively the mean and standard deviation from a set of data. The proposed thresholding process leads to a reduced dictionary

shown as follows:

$$\mathcal{C}^{sep^*} = \{C^{sep,l^*}\}_{l^*=1,\ldots,L^*}, \tag{4.10}$$

where $L^* < L$ and $l^* \in l$. Training data at each time instant k is then associated to a word such that $C_k^{sep} \in \mathcal{C}^{sep^*}$.

Observe that at a single time instant k, multimodal observations $\left(Z_k^m\right)$ can be interpreted in three different levels: GSs $\left(\tilde{X}_k^m\right)$, independent letters $\left(S_k^m\right)$ and words $\left(C_k^{sep}\right)$. This work proposes a hierarchical relationship among the aforementioned variables; where more codified variables determine the behavior of the less codified ones. Accordingly, Figure 4.2 displays the proposed C-DBN architecture employed by the S-A protocol for making inferences at a multilevel fashion over a bi-modal (m and m') case.

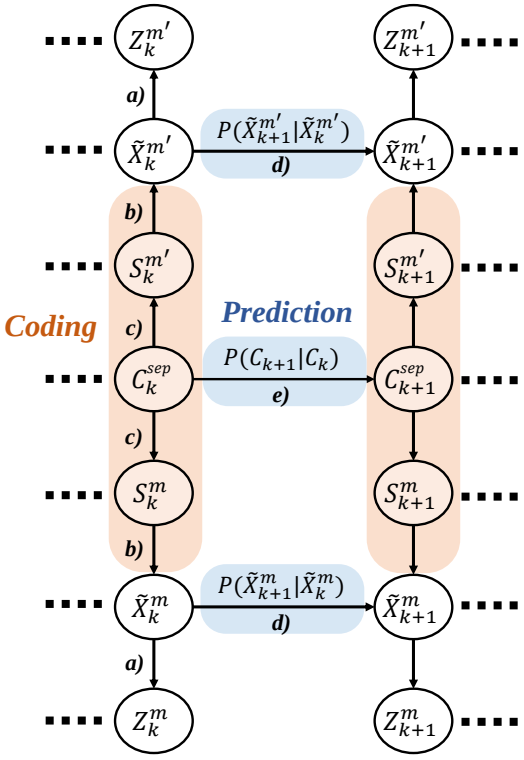


FIGURE 4.2: Proposed structure of C-DBN for the S-A. Links related to the coding of GSs into discrete variables are indicated in orange whereas prediction links are highlighted in blue

The C-DBN architecture employed by the S-A is shown in Figure 4.2. The coding stage in the S-A distinguishes between three types of hierarchical (multilevel) relationships at each time instant (see vertical arrows labeled as ***a)***, ***b)*** and ***c)*** in Figure 4.2). Each of these relationships is modeled as follows:

- Arrows labeled as ***a)*** relate measurements with GSs for each modality. By extending equation 4.3 to a multimodal framework, it is possible to write:

$$Z_k^m = X_k^m + \nu_k^m, \tag{4.11}$$

where ν_k^m encodes the observation noise produced by the sensors associated with the modality m. At each time instant, it is assumed that observations depend on GSs, i.e., $p(Z_k^m|\tilde{X}_k^m)$.

- Arrows labeled as ***b)*** relate GSs to vocabulary elements. Note that each modality cluster (letter) can be characterized by its mean and variance in the GS space. Accordingly, it is possible to determine the current cluster S_k^m by calculating the euclidean distance between $\tilde{X}_k^m$ and existing clusters $\mathcal{S}^m$ and selecting the closest one. At each instant k, it is assumed that GSs depend on the calculated letters, i.e., $p(\tilde{X}_k^m|S_k^m)$.

- Arrows labeled as ***c)*** relate vocabulary elements to a given word. As suggested by the proposed definition of words, once the letters of different modalities are calculated at a given time k, a word that combines them is automatically calculated based on a created dictionary (see Equation 4.10). It is assumed that activated letters of each modality depend on the active word at each time instant k, i.e., $p(S_k^m|C_k^{sep})$.

On the other hand, the C-DBN structure in Figure 4.2 has two types of horizontal arrows labeled as ***d)*** and ***e)*** corresponding to the prediction stage at continuous and discrete levels respectively. Both levels of inferences are discussed as follows:

- The arrows labeled as ***d)*** facilitate the estimation of each modality's GSs at a time $k+1$ given observations until the time k, i.e., $p(\tilde{X}_{k+1}^m|\tilde{X}_k^m)$. Such predictions at the continuous level depend linearly on the previous GSs such that:

$$\tilde{X}_{k+1}^m = A\tilde{X}_k^m + BU_{S_k^m} + \omega_k^m, \tag{4.12}$$

where the matrix A takes the state components of the GSs and makes null their time derivative components. U_{S_k} represents the calculated action of the GSs associated with the modality m at the time k and ω_k^m models the noise of the proposed dynamic model. Note that the change of GSs, $U_{S_k^m}$ depends on the discrete level variable S_k^m which in turn depends on the higher discrete

variable C_k^{sep} which encodes combined effect of module m and other modules $m' \neq m$.

- The arrow labeled as ***e)*** is responsible for predicting the combined state of multimodal data, i.e., a word, at a time instant $k+1$ given observations until the previous instant k, i.e., $p(C_{k+1}^{sep}|C_k^{sep})$. Since a word C_{k+1}^{sep} is a discrete variable resulting from a two-step clustering approach over GSs, its prediction is performed by taking into consideration a transition/stochastic matrix that encodes the probability of going from C_k^{sep} to any other word in $\mathcal{C}^{sep^*}$.

4.3.2 Joint approach (J-A)

Similar to the S-A, multi-sensory GSs are also clustered by the J-A in order to build a multilevel C-DBN architecture that facilitates the prediction of future instances. As depicted in Figure 4.3, the J-A clustering protocol consists of a single codification stage which groups directly multi-modal GSs into discrete variables.

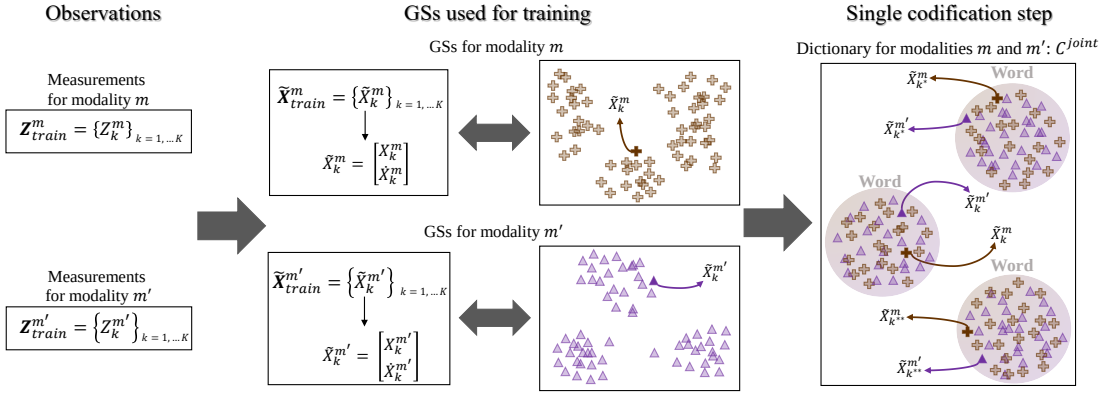


FIGURE 4.3: Codification process for generating discrete states in the J-A clustering protocol. Observations from bi-modal (m and m') information produce GSs that are directly grouped into clusters (words) that encode combined behaviors of the modalities is question. Note that k, k^* and k^{**} represent different time instants belonging to the training set.

By directly clustering multi-modal GSs, the J-A generates a dictionary $\mathcal{C}^{joint}$ that encodes a combined representation of involved GSs. Such a dictionary is defined as:

$$\mathcal{C}^{joint} = \{C^{joint,l}\}_{l=1,...,L}, \tag{4.13}$$

where $C^{joint,l}$ represents the l cluster obtained through an unsupervised clustering (GNG) algorithm over GSs. Similar to the S-A approach, $C^{joint,l}$ can be seen as

a word (discrete variable) containing mixed information of involved modalities. Note that the J-A does not take into consideration a vocabulary/letters for each modality, making the J-A less semantic than the S-A.

At each time instant, it is possible to represent the combined state of multi-modal data through the variable $C_k^{joint} \in \mathcal{C}^{joint}$. Such a variable is considered to influence the GSs that are responsible for the measured data. As explained previously, it is assumed that codified variables determine the behavior of the less codified ones. Accordingly, Figure 4.4 displays the proposed C-DBN architecture employed by the J-A protocol for a bi-modal (m and m') case. Two types of hierarchical relationships can be distinguished in Figure 4.4 corresponding to the vertical arrows labeled as ***a)*** and ***b)***. Both relationships are modeled as follows:

- Arrows labeled as ***a)*** have the same definition proposed in the S-A protocol. They relate measurements to multi-modal GSs through equation 4.11 allowing the calculation of $p(Z_k^m|\tilde{X}_k^m)$.
- Arrows labeled as ***b)*** relate GSs to dictionary elements (words). As mentioned previously, such a relationship is defined by an unsupervised clustering algorithm that takes as inputs all multimodal GSs and combines them into clusters as shown in Figure 4.3. Similar to the S-A, it is possible to calculate the mean and variance of the clustered components in each produced word in the J-A. In that way, it is possible to determine the current cluster C^{joint} by calculating the euclidean distance between involved $\tilde{X}_k^m$ and existing words $\mathcal{C}^{joint}$. It is assumed that multimodal GSs depend on the active word at a given time k, i.e., $p(\tilde{X}_k^m|C_k^{joint})$.

Similar to the S-A protocol, the DBN structure in Figure 4.4 has also two types horizontal arrows labeled as ***c)*** and ***d)*** associated with the prediction at continuous and discrete levels such that:

- The arrows labeled as ***c)*** enable the estimation of modality's $p(\tilde{X}_{k+1}^m|\tilde{X}_k^m)$. Such a probability can be written as:

$$\tilde{X}_{k+1}^m = A\tilde{X}_k^m + BU_{C_k^{joint}} + \omega_k^m, \tag{4.14}$$

  where all variables have a direct correspondence to the dynamical model discussed in equation 4.12 except for the term $U_{C_k^{joint}}$ which depends directly from the created words $\mathcal{C}^{joint}$ (see equation 4.13).

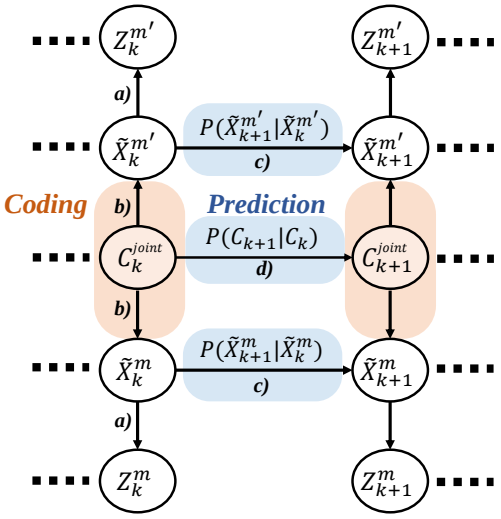


FIGURE 4.4: Proposed structure of C-DBN for the J-A. Links related to the coding of GSs into discrete variables are indicated in orange whereas prediction links are highlighted in blue

- The arrow labeled as ***e)*** encodes the prediction of a future word given the current one, i.e., $p(C_{k+1}^{joint}|C_k^{joint})$. Similar to S-A, the prediction is executed thought a transition/stochastic matrix that encodes the probability of going from C_k^{joint} to any other word in $\mathcal{C}^{joint}$.

As can be seen by comparing the diagrams in Figure 4.1 and Figure 4.3, the main difference between both clustering protocols consists of an additional semantic information generated in the first-step of the S-A protocol and that does not exist in the J-A. Such a difference in the coding of GSs between both protocols can be translated into slightly different DBN architectures, compare Figure 4.2 and Figure 4.4. Despite the differences of coding GSs between S-A and J-A protocols, their dictionaries, $\mathcal{C}^{sep^*}$ and $\mathcal{C}^{joint}$, both contain combined information of the involved modalities and facilitate the predictions of future states in a hierarchical probabilistic fashion.

As described through this section, observations of multimodal data Z_{train}^{m} is employed by two different clustering protocols (S-A and J-A) in order to build predictive models that will be employed for evaluating/analyzing new (unseen) multimodal measurements. Accordingly, C-DBNs presented in Figure 4.2 and Figure 4.4 are not only employed for representing multimodal observations but for making inferences when they are used over a set of testing data.

4.4 Testing phase: State estimation and abnormality detection

For making inferences over testing data, an improved version of the MJPF (section 3.2.3.1) is presented. The proposed MJPF takes into consideration a PF where each particle models and predicts the dynamics of the multimodal GSs by using KF. In other words, our MJPF makes inferences about multimodal GSs by considering interactions among them. Such interactions are modeled as discrete variables (words: C_k^{sep} and C_k^{joint}) through a PF approach coupled with a bank of KFs.

The inputs of the MJPF are similar for both clustering protocols (S-A and J-A) and are described as follows:

- ***Mean value of clusters:*** It consists of the mean value of GSs grouped in clusters. In the S-A, it corresponds to the average value of the GSs inside each letter $S^{n_m} \in \mathcal{S}^m$ (see Equation 4.8), and later maps them into words (see Equation 4.9). Accordingly, let us represent the mean values of grouped GSs in the S-A dictionary as:

$$\overline{\mathcal{C}^{sep}} = \{\overline{C^{sep,l}}\}_{l=1,\dots,L}, \tag{4.15}$$

  where $\overline{C^{sep,l}} = \{\overline{S^{n_m}}\}_{m=1,\dots,M}$. $\{\overline{S^{n_m}}\}$ represents the average value of GSs associated to the letter S^{n_m}. Recall that n_m indexes the letters associated with the vocabulary/modality m.

  For the J-A case, let $\overline{C^{joint,l}}$ be the set of averaged GSs related to words' clusters as:

$$\overline{\mathcal{C}^{joint}} = \{\overline{C^{joint,l}}\}_{l=1,\dots,L}. \tag{4.16}$$

- ***Covariance matrix of clusters:*** It consists of the covariance matrix calculated over GSs of clusters. In the S-A, it plays the role of extracting the covariance based on GSs in each letter and later maps them into words. Covariance matrices of grouped GSs in the S-A dictionary can be written as follows:

$$\mathcal{C}_{cov}^{sep} = \left\{ cov\left(C^{sep,l}\right) \right\}_{l=1,\dots,L}, \tag{4.17}$$

where $cov(\cdot)$ is a function that extracts the covariance matrix from the input data. As discussed previously, each word is composed of multiple letters such that $C^{sep,l} = \{S^{n_m}\}_{m=1,\dots,M}$.

For the J-A case, the set of covariance matrices related to the multimodal GSs codified in words can be written as follows:

$$\mathcal{C}_{cov}^{joint} = \left\{ cov\left(C^{joint,l}\right) \right\}_{l=1,\dots,L}. \tag{4.18}$$

- ***Radius of acceptance:*** It consists of a boundary/limit value that is employed to indicate the validity of built models on the testing data. Specifically, each word has a radius of acceptance that together with its mean value (see Equation 4.15) define where created models are valid. In the S-A, the set of radius of acceptances is represented as:

$$\mathcal{C}_{rad}^{sep} = \left\{ 3\, tr\left(\sqrt{cov(C^{sep,l})} \right) \right\}_{l=1,\dots,L}, \tag{4.19}$$

where $tr(\cdot)$ represents the trace operation and the square root operation is applied to all elements of the covariance matrix. The idea behind the proposed radius of acceptance lies on the 99.7 rule, which defines the validity of models based on a maximum deviation from the mean by 3 times the standard deviation of data.

For the J-A case, the set of radius of acceptances of multimodal words can be written as follows:

$$\mathcal{C}_{rad}^{sep} = \left\{ 3\, tr\left(\sqrt{cov(C^{joint,l})} \right) \right\}_{l=1,\dots,L}, \tag{4.20}$$

- ***Transition matrix of the dictionary:*** In both clustering protocols, S-A and J-A, transition (stochastic) matrices at the level of words are calculated by employing a frequentist interpretation of the probability of going from a word in a given time k to another one at the time $k+1$. Accordingly, training data is used to count the number of jumps between multimodal words, so that the transition matrix can be calculated for S-A and J-A protocols based on the frequency of going from one word to another. Let transition matrices be written as $\mathcal{P}^{sep}$ and $\mathcal{P}^{joint}$ for S-A and J-A protocols respectively. Both

matrices facilitate the way to calculate the probabilities $p(C_{k+1}^{sep}|C_k^{sep})$ and $p(C_{k+1}^{joint}|C_k^{joint})$.

- ***Testing data**:* It consists of data series of multimodal observations Z_{test}^m, employed for evaluating the proposed method's capability at inferring future GSs (prediction purposes) and detecting abnormalities with different clustering protocols.

The logic of the MJPF is the same for evaluating both clustering protocols. Algorithm 3 shows the different steps of the MPJF and evidences the way by which the MJPF uses the aforementioned clusters' information for prediction and detection of abnormalities.

The MJPF can be divided into five main steps that are executed at each time instant k:

- ***Word calculation:*** where current discrete variable(s) are calculated, i.e., ongoing word and letters in the S-A case (see Figure 4.5).
- ***Prediction step:*** it consists in estimating following continuous (GSs) and discrete (word) random variables in $k+1$ (see Figure 4.5).
- ***Abnormality detection:*** where the differences between predictions and evidence are calculated (additionally see next section 4.4.1).
- ***Particle resampling:*** abnormalities are used to measure particles' weights. Particles are then redistributed based on the calculated weights (see Figure 4.6).
- ***Update step:*** GSs are updated based on the present measurement.

Algorithm 3 Markov Jump Particle Filter

Input:

$\bar{\mathcal{C}}$: Average GSs values associated with words

$\mathcal{C}_{cov}$: Covariance matrices associated with words

$\mathcal{C}_{rad}$: Acceptance radius associated with words

$\mathcal{P}$: Transition matrix of words

N : Total number of particles

$\boldsymbol{Z}_{testing}$: Testing multimodal measurements

K : Total number of testing measurements

Output:

$\{\boldsymbol{\theta}, \boldsymbol{\phi}\}$: Abnormality signals

procedure PREDICTION OF GSs BASED ON
MULTI-MODAL DATA
    Initialize current time and particle: $k = 1$, $n = 1$
    Initialize $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ as empty vectors
`Multilevel prediction:`
    if $k == 1$ **then**
        $\tilde{X}_k^{m,(n)} \leftarrow$ GSs of the particle n at time k
                based on $Z_k \in \boldsymbol{Z}_{testing}$
    $C_k^{(n)} \leftarrow$ Current word based on $\bar{\mathcal{C}}$, $\mathcal{C}_{rad}$ and $\tilde{X}_k^{m,(n)}$
    $\hat{C}_{k+1|k}^{(n)} \leftarrow$ Word prediction based on $\mathcal{P}$ and $C_k^{(n)}$
    $\hat{X}_{k+1|k}^{m,(n)} \leftarrow$ GSs prediction based on $C_k^{(n)}$ and $\tilde{X}_k^{m,(n)}$
    if $n == N$ **then**
        $k := k + 1$
        $\{\theta_k^m, \phi_k^m\} \leftarrow$ Abnormality measurements
        based on $Z_k \in \boldsymbol{Z}_{testing}$, $\hat{X}_{k+1|k}^{m,(n)}$, $\bar{\mathcal{C}}$ and $\mathcal{C}_{cov}$
        $\{\boldsymbol{\theta}, \boldsymbol{\phi}\} \leftarrow$ Append $\{\theta_k^m, \phi_k^m\}$ respectively
        Resampling of all particles' GSs based on
        based on abnormalities $\{\theta_k^m, \phi_k^m\}$
        $\tilde{X}_k^{m,(\eta)} \leftarrow$ All resampled particles are updated,
        where $\eta = 1, \ldots, N$
        $n = 0$
    $n := n + 1$
    if $k < K$ **then**
        Go to `Multilevel prediction`
    else
        return $\{\boldsymbol{\theta}, \boldsymbol{\phi}\}$

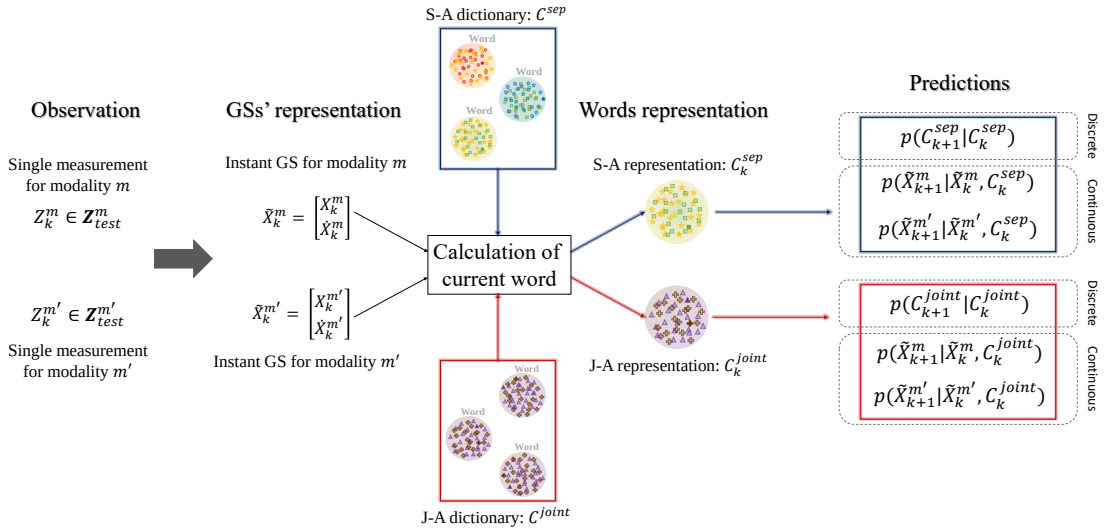


FIGURE 4.5: Application of the MJPF for prediction purposes. Measurements from a single time instant (k) are employed to calculate the current GSs. Dictionaries from S-A and J-A clustering protocols are then used to calculate the current word. Consequently, by employing transition/stochastic matrices and continuous dynamic models (see Equation 4.12 and Equation 4.14). It is possible to predict/estimate the word and GSs of the next time instant $(k+1)$.

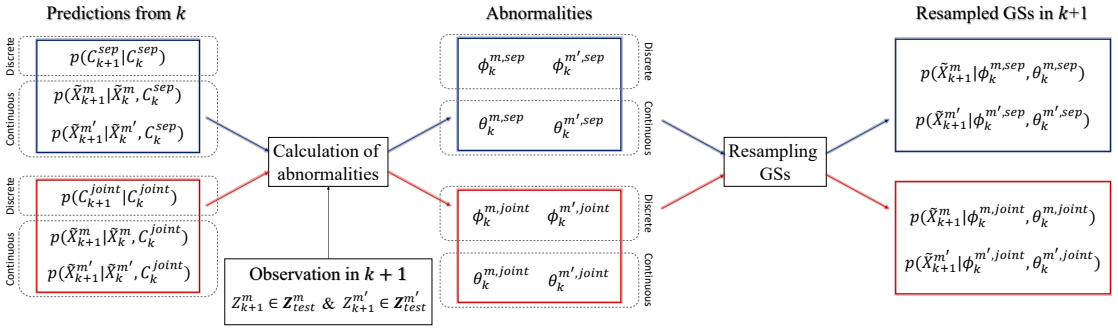


FIGURE 4.6: Detection of abnormalities and resampling using the MJPF. Previously calculated predictions from the instant k and current measurements $k+1$ (see Figure 4.5) are employed to calculate the deviations of our predictions w.r.t the evidence, i.e., abnormalities. Subsequently, abnormality measurements are employed to individuate the particles that predict better such that new particles are initialized based on them.

4.4.1 Abnormality detection

Broadly speaking, abnormalities can be defined as a behavior pattern that has not been observed and learned before [90, 98]. Abnormalities can be then measured as the difference between predictions (coming from learned models) of future time

instances and the actual evidence. Accordingly, two types of abnormalities are here considered to evaluate proposed clustering protocols (S-A and J-A): one associated with the continuous level (see Equation 4.21) and other related to the discrete levels (see Equation 4.22) as pointed out in section 3.2.4.3.

$$\begin{aligned}\theta_{k+1}^{m,sep} &= D_B\Big(p(\tilde{X}_{k+1}^m|\tilde{X}_k^m(C_{k+1}^{sep})), p(Z_{k+1}^m|\tilde{X}_{k+1}^m)\Big)\\ \theta_{k+1}^{m,joint} &= D_B\Big(p(\tilde{X}_{k+1}^m|\tilde{X}_k^m(C_{k+1}^{joint})), p(Z_{k+1}^m|\tilde{X}_{k+1}^m)\Big)\end{aligned} \tag{4.21}$$

$$\begin{aligned}\phi_{k+1}^{m,sep} &= D_B\Big(p(\tilde{X}_{k+1}^m|\tilde{X}_k^m(C_{k+1}^{sep})), p(\tilde{X}_{k+1}^m|C_{k+1}^{sep})\Big)\\ \phi_{k+1}^{m,joint} &= D_B\Big(p(\tilde{X}_{k+1}^m|\tilde{X}_k^m(C_{k+1}^{joint})), p(\tilde{X}_{k+1}^m|C_{k+1}^{joint})\Big)\end{aligned} \tag{4.22}$$

$D_B(p,q)$ represents the Bhattacharyya distance between the probability distributions p and q as presented in section 2.3.3.2. Accordingly, Equation 4.21 encodes the difference between the predicted GSs of the instant $k+1$ and updated GSs at $k+1$, i.e., after obtaining observations at $k+1$. Additionally, Equation 4.22 represents the difference between the predicted GSs of the instant $k+1$ and the word calculated at $k+1$. Accordingly, Bhattacharyya distances in Equation 4.21 encode abnormalities at the continuous level whereas equation 4.22 encode them at the discrete level. As summarized in Figure 4.6, predictions obtained at each time instant are employed to calculate abnormalities that in turn are used to resample particles' GSs of the PF.

4.5 Employed dataset

As shown in Figure 1.2, an agent can perceive and distinguish two types of sensory information related to:

i) Its own internal states by proprioceptive sensors.

ii) Its surroundings by exteroceptive sensors.

Accordingly, SA in the artificial agent is here modeled as a multi-sensory problem where internal and external perceptions are employed to make inferences of future agent's states based on models that are learned. Coupling of exteroceptive and proprioceptive models arises from the need of identifying causalities/interactions

between multi-sensory data perceived by an artificial agent. By coupling the exteroceptive and proprioceptive models, it is possible to build a model that takes into consideration a contextual viewpoint for making inferences about future perceived information.

The dataset presented in Section 2.6.1.1 is used to test our proposed approaches. Accordingly, multi-modal data from the perimeter monitoring scenario is here employed as input for the training phase (see section 4.3). Additionally, the avoidance maneuver scenario is utilized as testing data for performing inferences of future GSs and words (see section 4.4) and detecting abnormalities (see section 4.4).

Bi-modal data from a real vehicle, i.e., odometry modality (exteroceptive data) which contains positional data mapped into Cartesian coordinates (x,y) and control modality (proprioceptive data) which consists of information related to the controls of the vehicle, i.e., steering angle s and rotors velocity v are considered to evaluate and compare the performances of S-A and J-A protocols under different compression levels.

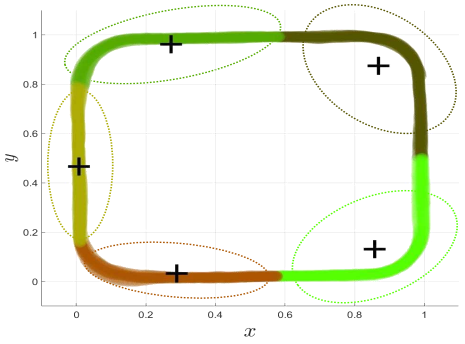
4.6 Fair comparison setup

For guaranteeing a fair comparison between the S-A and J-A protocols, we proposed a fairness criterion based on the number of dictionary elements obtained in the training phase of both protocols. In the proposed methodology, the number of dictionary elements can be seen as a measurement of data compression. Consistently, large dictionaries produce a high data compression whereas short dictionaries generate a low compression rate. As it is known, short dictionaries that group information into meaningful clusters are preferred over redundant enormous dictionaries. Nonetheless, finding the correct dictionary size for a particular purpose, e.g., abnormality detection or prediction, is not a simple task. Accordingly, this work considers three compression levels:

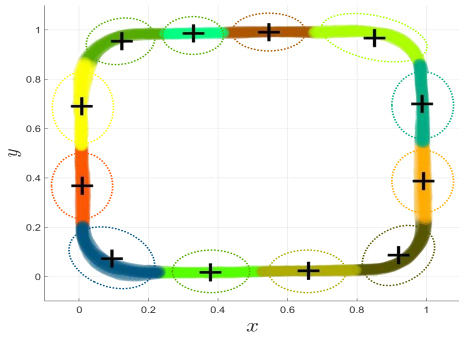
- ***Under-clustering.***
- ***Mid-clustering.***
- ***Over-clustering.***

Figure 4.7 depicts the clusters of positional data from the perimeter monitoring scenario related to the proposed three-level compression process.

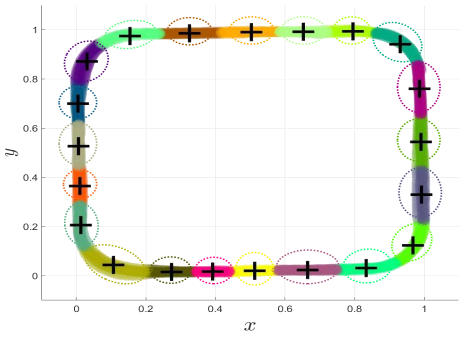
From Figure 4.7, it is possible to see that different compression levels that facilitate a distinct GS coverage of clusters. To fairly compare S-A and J-A clustering protocols, each compression level guarantees the same number of discrete elements in both approaches. In other words, the dictionaries $\mathcal{C}^{joint}$ and $\mathcal{C}^{sep^*}$ have the same number of elements (words) at each level.



(A) Under-clustering



(B) Mid-clustering



(C) Over-clustering

FIGURE 4.7: Three different clustering compression levels applied on positional data. Crosses indicate the average point of clusters. Positional data associated with each cluster is colored differently. Additionally, it is considered an ellipse that covers the area where the majority of data is concentrated in each cluster.

The fairness of comparing S-A and J-A protocols lies on considering the same number of discrete variables at their highest inference level. Since the instantaneous values of other random variables, i.e., letters, GSs and observations depend on the current word, fixing the size of dictionaries at each compression level becomes a choice when comparing S-A and J-A protocols. This work proposes a set of measurements to evaluate the quality of obtained clusters (*offline evaluation*) and their performance at predicting/detecting abnormalities (*online evaluation*). Our work evaluates two main concepts in the obtained clusters:

- ***Coverage of GS space:*** it refers to the capacity of the clusters to cover large areas of the GSs. A large coverage of GSs helps AAs to reduce its uncertainty when making predictions in areas that have not seen previously.
- ***Model precision:*** it refers to the clusters' prediction capabilities, which depends on the compactness of their first time derivative information. Clusters with high compactness in their first time derivatives tend to produce more stable predictions since the linearity in the dynamic models of the bank of KFs is preserved.

4.6.1 Offline evaluation of clusters

To evaluate the quality of produced clusters in the training phase, four measurements are proposed:

- ***Variance of GSs' components.*** Clusters that occupy large areas of the state space (X_k^m information) while preserving compact first time derivative components ($\dot{X}_k^m$ information) are preferred. Accordingly, the ideal clusters will exhibit a high variance in their state spaces and a low variance in their first derivative components, leading to precise linear dynamical models (where models in Equation 4.12 and Equation 4.14 are valid) on extensive areas (assuring a large coverage of the GS).
- ***Words' entropy.*** As explained previously, by considering a frequentest approach over the series of words associated with GSs belonging to the training data, it is possible to generate the transition matrix $\mathcal{P}$. Probabilities encoded

in such a matrix are employed to calculate the entropy of clusters. The information entropy of $\mathcal{P}$ is calculated by using the next expression:

$$\mathrm{S} = -\sum_{i} \mathrm{p}_i \log \mathrm{p}_i, \tag{4.23}$$

where $\mathrm{p}_i \in \mathcal{P}$ and i indexes the cells of the transition matrix. A low entropy is preferred since it indicates a more precise/certain discrete dynamics when predicting future words.

- ***Number of connected words.*** By looking at the generated clusters as a graph structure where each node corresponds a given multi-modal cluster and the presence/absence of links encode the closeness/remoteness among other clusters. It is possible to consider the total number of connections as measurement of clusters' quality. A low number of links is preferred since it is expected that each multi-modal cluster (word) represents a unique concept separate from others.

- ***Training time.*** The time for generating multi-modal clusters (words) is measured and compared between S-A and J-A protocols for different compression levels.

4.6.2 Online evaluation of clusters

As shown in Figure 4.5 and Figure 4.6, by applying the MJPF on test data, it is possible to obtain a series of predictions and abnormality signals which are employed to evaluate the quality of clusters in the three following ways:

- ***ROC curve properties.*** By considering testing data that carries known normal (previously seen in the training data) and abnormal (new experiences) behaviors, it is possible to use such a ground truth information to measure the performance of proposed models. Consequently, ROC curves can be obtained and the Area Under Curve (AUC) and Accuracy (ACC) measurements can be calculated and employed to evaluate the performance of clusters in online applications. High AUC and ACC values are preferred since they encode the MPJF capabilities of detecting abnormal/normal behaviors precisely.

- ***Local prediction error.*** It is considered a normalized version of the error between predictions and measurements, i.e., innovations. Accordingly, at each time instant, for each modality m, it is calculated the following error measurement:

$$\Delta X_k^m = \left| \frac{X_{k|k-1}^m - Z_k}{X_{k|k-1}^m + Z_k} \right|, \tag{4.24}$$

where $X_{k|k-1}^m$ represents the prediction of the state X at the time instant k given observations until the time $k-1$. Z_k^m is the measurement related to the modality m at time k. $X_{k|k-1}^m$ is the prediction of the state space at the time k given observations until time $k+1$ for the modality m.

The fair comparison between errors produced by the two proposed clustering protocols (S-A and J-A) is then performed for each multisensorial as follows:

$$\mathcal{E}_k^m = \frac{\Delta X_k^{m,(sep)} - \Delta X_k^{m,(joint)}}{\Delta X_k^{m,(sep)} + \Delta X_k^{m,(joint)}}, \tag{4.25}$$

where $\Delta X_k^{m,(sep)}$ and $\Delta X_k^{m,(joint)}$ correspond to normalized error measurement shown in Equation 4.24 related to the S-A and J-A protocols respectively. Since multisensorial data is considered, at each time instant a final measurement E_k is obtained by summing all error components in Equation 4.25 for the different modalities such that:

$$E_k = \sum_{m=1}^{M} \sum_{d_m=1}^{D_m} \mathcal{E}_k^{m,d_m}, \tag{4.26}$$

where $\mathcal{E}_k^{m,d_m} \in \mathcal{E}_k^m$ represents the error associated to the d_m component of the modality m. As discussed in section 4.2, M represents the total number of modalities and D_m is the number of state space dimensions in the modality m.

The expression in Equation 4.26 encodes a comparison between the S-A and J-A predictions. By introducing a threshold $\lambda = \bar{\boldsymbol{E}}$, where $\boldsymbol{E} = \{E_1, E_2, \dots E_K\}$ that considers possible oscillations in the comparison of errors, it is possible to detect $E_k < -\lambda$ as instances where the S-A predicts with considerably less error than J-A. On the other hand, $E_k > \lambda$ refers to instances where the J-A produces less errors than S-A. Additionally, when $-\lambda < E_k < \lambda$, it refers to cases where there is no significant difference in the predictive performance between both clustering protocols. Accordingly,

let $\%E^{sep}$ and $\%E^{join}$ be respectively the perceptual rate by which the S-A and J-A were favored, i.e., $E_k < -\lambda$ and $E_k > \lambda$ respectively. Additionally, let $\%E^{N/A}$ be the number of times where none protocol is favored, i.e., $-\lambda < E_k < \lambda$ such that $\%E^{sep} + \%E^{join} + \%E^{N/A} = 100\%$.

- ***Model coverage.*** In this part, we introduce a measurement that evaluates the GS coverage of proposed models. The main idea consists of measuring how a set of clusters deals with data that differs substantially from the training set. Accordingly, we take into consideration the amount of times that testing data falls outside the radius of acceptance (see Equation 4.19 and Equation 4.20 for S-A and J-A respectively). Each time a data sample goes out the acceptance region, the MPJF executes a random estimation of the next discrete word producing uncertain predictions in the following time instants. As can be intuited, going out from the model's radius of acceptance generates flaws in its predictions and affects its robustness when facing observations that differ substantially from the training data.

  Since a PF approach is considered, it is used to calculate the percentage of particles that fall outside the acceptance region $\%N_{out,k}$ when predicting at each time instant k. When a large number of particles goes outside the the radius of acceptance, i.e., high $\%N_{out,k}$, the model's predictions are not reliable since states cannot be fully explained by previous learned dynamics. In this sense, low $\%N_{out,k}$ values are preferred over larger ones since they indicate a larger GS coverage of the model. S-A and J-A protocols generate at each instant a percentage of particles going out the model, $\%N^{sep}_{out,k}$ and $\%N^{join}_{out,k}$ respectively. By setting K as the total number of testing observations, the next step is to compare the correspondent K elements of $\%N^{sep}_{out,k}$ and $\%N^{join}_{out,k}$. Accordingly, it is identified the total of $K^{\dagger}$ instances where $\%N^{sep}_{out,k}$ and $\%N^{join}_{out,k}$ are different where $K^{\dagger} \leq K$. Subsequently, a voting process is performed where ν^{sep} and ν^{join} are defined as the number of instances belonging to $K^{\dagger}$ where S-A and J-A protocols respectively presented a lower percentage of particles that went out of the model. Two final variables $\%\mathcal{V}^{sep}$ and $\%\mathcal{V}^{join}$ are employed for evaluating the number of votes that favor S-A and J-A respectively regarding their capabilities for coveraging the state space, such that:

$$\%\mathcal{V}^{sep} = 100\frac{\nu^{sep}}{K^{\dagger}}, \quad \%\mathcal{V}^{join} = 100\frac{\nu^{join}}{K^{\dagger}}, \tag{4.27}$$

where $\%\mathcal{V}^{sep} + \%\mathcal{V}^{join} = 100\%$.

4.7 Experimental results

Our main purpose is to examine the advantages and disadvantages of such two clustering protocols represented by different DBN architectures (see Figure 4.2 and Figure 4.4). As explained through this chapter, our method can be split into two main phases namely training and testing. Accordingly, properties of learned dictionaries based on S-A and J-A protocols are compared in section 4.7.1. Subsequently, trained models are used to predict following time instances in testing data, abnormality signals and performance measurements that compare S-A and J-A protocols are provided in section 4.7.2.

4.7.1 Training phase

By training S-A and J-A protocols at the three proposed compression levels (Under/Mid/Over-clustering), it is possible to evaluate and compare produced clusters based on the variances in their components. Table 4.1 shows the variances of each GS component of performed clusters that include the usage of odometry and control modalities from the vehicle (see section 4.5).

Bold values in Table 4.1 indicate a significant out-performance when comparing S-A and J-A protocols at the different compression levels. It can be seen that the J-A produces more compressed first time derivatives (lower values in $\sigma^2_{\dot{x}}$, $\sigma^2_{\dot{y}}$, $\sigma^2_{\dot{s}}$, $\sigma^2_{\dot{v}}$) than the S-A which leads to more precise dynamical models. Nonetheless, the S-A produces models that cover larger areas in the odometry data (higher values in σ^2_x, σ^2_y) that represents an advantage when dealing with abnormal data. It can be seen also that statistical information from the state space of the control module does not provide any preference towards S-A nor J-A. The latter is due to the fact that it tends to carry high levels of noise, specially the velocity of motor v which does not exhibit any precise pattern when the vehicle performs the proposed tasks.

Table 4.2 shows different cluster metrics employed for evaluating the quality of S-A and J-A protocols at different compression levels. As explained in section 4.6, the proposed fair comparison guarantees a similar number of clusters N at each compression level for S-A and J-A. The training time of the J-A is significantly lower

than the S-A. Such result is expected since the SA uses a two-step clustering and J-A just a single one. The number of graph connections N_{conn} and the entropy S (see Equation 4.23) show a similar behavior. They favor the J-A when performing the under-clustering compression and the S-A in the over-clustering case. Additionally, no special favoritism is found when considering the middle-clustering compression. The evidence suggests that as the number of clusters increases, the S-A presents more precise discrete transitions. The latter is supported by the larger values in state space variances (see σ_x^2, σ_y^2) obtained with the S-A which suggests a better GS coverage.

	Variable	**Under-clustering**		**Middle-clustering**		**Over-clustering**	
		Sep.	*Joint*	*Sep.*	*Joint*	*Sep.*	*Joint*
Odometry	σ_x^2	**5.0639**	1.1688	**1.9606**	0.7078	**1.1335**	0.5091
	σ_y^2	**3.8804**	1.0267	**1.9297**	0.5534	**0.9758**	0.4008
	$\sigma_{\dot{x}}^2$	0.0655	**0.0391**	0.0488	**0.0334**	0.0418	**0.0285**
	$\sigma_{\dot{y}}^2$	0.0811	**0.0409**	0.0584	**0.0322**	0.0422	**0.0269**
Control	σ_s^2	2.0197	2.0899	1.2982	1.1885	1.1138	0.9059
	σ_v^2	0.0334	0.0377	0.0184	0.0201	0.0138	0.0123
	$\sigma_{\dot{s}}^2$	0.3995	**0.2088**	0.3584	**0.1339**	0.3420	**0.1026**
	$\sigma_{\dot{v}}^2$	0.0022	**0.011**	0.0021	**0.0007**	0.0020	**0.0006**

TABLE 4.1: Comparison of S-A and J-A cluster component variances for different compression levels. Bold values evidence when a clustering approach significantly outperforms the other one.

Variable	**Under-clustering**		**Middle-clustering**		**Over-clustering**	
	Sep.	*Joint*	*Sep.*	*Joint*	*Sep.*	*Joint*
N	23	23	83	82	160	162
N_{conn}	70	**49**	218	206	**372**	412
S	0.0844	**0.0586**	0.0869	0.0921	**0.0823**	0.1098
T_{train} (sec)	2.0554	**0.4482**	2.1093	**0.7871**	2.1026	**1.6962**

TABLE 4.2: Comparison of S-A and J-A cluster properties: Number of nodes (N), number of node connections (N_{conn}), entropy (S) and training time (T_{train}) for different compression levels. Bold values evidence when a clustering approach significantly outperforms the other one.

4.7.2 Testing phase

A MJPF is employed for both clustering protocols to make inferences about future of multi-modal information. The different measurement for evaluating the performance of S-A and J-A protocols at different compression levels are displayed in Table 4.3. It is important to mentioned that all testing experiments consider the same number of particles (40 particle in this case). Such a number is chosen through a hyper-parameter tuning study based on random search of the number of particles where both approaches present high performances without compromising the computational testing time.

	Variable (%)	**Under-clustering**		**Middle-clustering**		**Over-clustering**	
		Sep.	*Joint*	*Sep.*	*Joint*	*Sep.*	*Joint*
Odometry	AUC_θ	61.20	**80.57**	78.80	**94.78**	91.33	95.33
	AUC_ϕ	66.07	**88.67**	91.05	93.92	92.22	91.84
	ACC_θ	84.71	84.71	85.39	**91.18**	90.77	93.11
	ACC_ϕ	86.50	**91.73**	93.93	92.83	94.62	93.25
Control	AUC_θ	87.48	85.11	**87.64**	81.46	**89.93**	79.28
	AUC_ϕ	86.33	91.02	88.51	87.75	90.59	86.44
	ACC_θ	92.28	89.94	**92.42**	85.67	**92.01**	86.08
	ACC_ϕ	93.80	93.38	94.90	93.93	94.21	93.38
	$E^{sep/join}$	24.27	**47.03**	37.51	**47.45**	41.65	**49.93**
	$\mathcal{V}^{sep/join}$	**87.28**	12.72	**66.90**	33.10	**57.42**	42.58

TABLE 4.3: Evaluation and comparison of S-A and J-A protocols based on performance measurements (see section 4.6). Bold values evidence when a clustering approach significantly outperforms the other one.

As explained previously, both S-A and J-A clustering protocols produce two types of abnormalities (for the continuous and discrete level) at each time instant for each modality (see Equation 4.21 and Equation 4.22). Consequently, for evaluating the

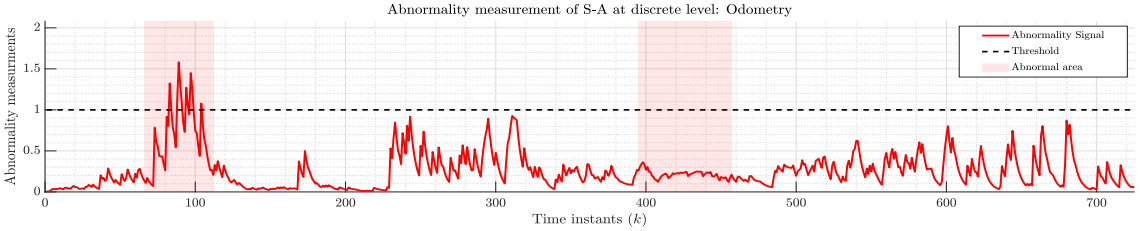
models' capabilities at detecting abnormalities and predicting future dynamics accurately, the AUC and ACC are calculated from ROC curves built based on proposed abnormality signals. Table 4.3 reports the AUC and ACC related to the abnormality signals taking into consideration that AUC_θ and ACC_θ are related to continuous abnormality signals (see Equation 4.21) whereas AUC_ϕ and ACC_ϕ to discrete ones (see Equation 4.22).

From Table 4.3, it is possible to see how AUC and ACC measurements favor the J-A at under and middle clustering compression levels when inferring odometry information. Nonetheless, both protocols present similar performances at the over-clustering level when inferring odometry data. On the other hand, the AUC and ACC measurement also indicate that the S-A protocol present a slightly better performance than the J-A when inferring information related to the control modality. Overall, the inference capabilities measured by AUC and ACC favor the J-A over the S-A.

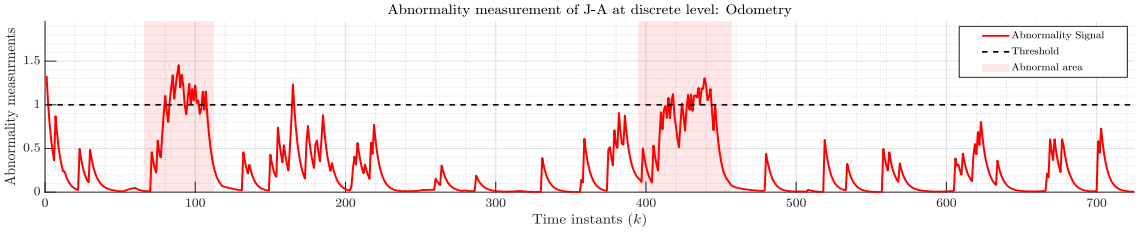
The local prediction error and the model coverage of S-A and J-A protocols are compared in the last two rows of Table 4.3 for the different compression levels. As can be seen, the local error measurements, i.e., $\%E^{sep}$ and $\%E^{sep}$, favor the J-A protocol over the S-A. This supports the overall result obtained from AUC and ACC measurements, reassuring that the J-A has better predicting capabilities than the S-A. Such an affirmation is also supported by the higher precision of dynamical models, see lower variances of first time derivative components in Table 4.1. On the other hand, the coverage of models clearly favor the S-A protocol regardless the compression level. This is supported by the S-A's higher variances in the positional state space in Table 4.1. Note that the highest performances of AUC and ACC together with the closest gap between S-A and J-A protocols regarding local prediction errors and model coverage is found with the over-clustering compression level.

One of the goals of the proposed approach is to find a multimodal clustering that produces a low number of dictionary elements (words) and that is capable of predicting future time instances accurately. A dictionary composed of less words leads to a lower number of available continuous models, decreasing the complexity of the testing phase when making inferences at high hierarchical levels. Consequently, results obtained with low and high number of words, i.e., under-clustering and over-clustering compression levels, are discussed in detailed as follows:

- ***Under-clustering results***. Some results that compare the S-A and J-A protocols when using a low number of words are selected to display the differences and capabilities of both protocols. Figure 4.8a and Figure 4.8b show the discrete abnormality signals from the odometry modality in S-A and J-A respectively. Abnormality signals presented in this work (see also Figure 4.12 and Figure 4.13) are normalized based on the performance of the training data such that values higher than 1 are considered as anomalies. Note that the AUC associated with the odometry abnormality signals, i.e., AUC_{ϕ}, present a big performance gap between S-A (66.07%) and J-A (88.67%) protocols. As can be inferred from Figure 4.8, abnormalities from S-A detect precisely the first abnormality (around $k = 100$) but not second one (around $k = 430$). On the other hand, by employing the J-A, both abnormalities are correctly detected leading to a considerable difference in the performance prediction of both protocols.



(A) Abnormality signal based on the odometry information in the S-A.



(B) Abnormality signal based on the odometry information in the J-A.

FIGURE 4.8: Discrete level abnormality signals (see Equation 4.22) of S-A and J-A protocols at the under-clustering compression level for odometry information. Ground truth abnormality regions are indicated as red background rectangles.

The accuracy of predictions done by S-A and J-A can be evaluated through the differences of local multi-modal errors as the stated in Equation 4.26. Figure 4.9 shows how the proposed measured error behaves through time. The range where no particular protocol presents better prediction performance, i.e., $[-\lambda, \lambda]$, is represented as a red region. Yellow and green areas encode situations in which J-A and S-A protocols outperforms the other one

respectively. As shown in Figure 4.9, the major part of the generated signal favors the J-A protocol. For a clearer visual comparison between the

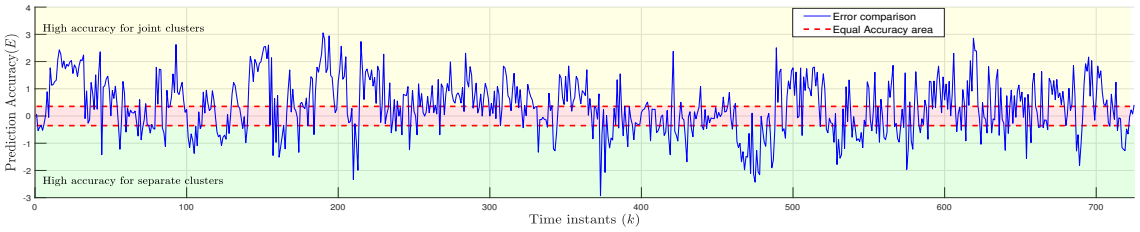


FIGURE 4.9: Local comparative error signal based on S-A and J-A prediction information, see Equation 4.26. The comparative error signal on the yellow/-green area, it indicates a more accurate prediction of the J-A/S-A respectively. When the signal goes on the area no particular protocol predicts better than the other.

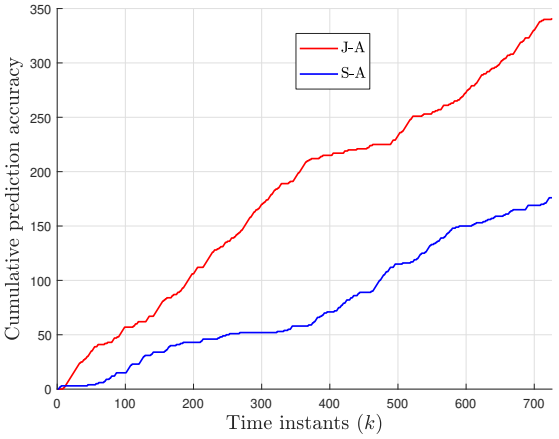


FIGURE 4.10: Cumulative prediction accuracy for S-A and J-A clustering protocol. Each time a protocol predicts better than the other; its score is incremented.

predicting errors produced by the S-A and J-A at the under-clustering compression level, the cumulative prediction accuracy of each protocol is plotted in Figure 4.10. The cumulative accuracy for both protocols is initialized as 0. As time (k) increments, the E_k defines whether the J-A or S-A should be incremented in a unit. Accordingly, when $E_k > \lambda$, the J-A signal (red line in Figure 4.10) is increased. Similarly, when $E_k < \lambda$, the S-A signal (blue line in Figure 4.10) is incremented. From those signals, it is evident that the J-A protocol outperforms the S-A when predicting multi-modal information accurately.

The coverage of S-A and J-A models at the under-clustering compression levels is also compared. Accordingly, Figure 4.11 displays the percentage of particles that go out the model's radius of acceptance at each time instant for

S-A and J-A, i.e., $\%N^{sep}_{out,k}$ and $\%N^{joint}_{out,k}$ respectively. As can be observed in Figure 4.10, the J-A protocol generates more particles out of the model than the S-A, showing an advantage in using the S-A in terms of model coverage. Note that in ground truth abnormality regions (red background rectangles), all particles of both clustering approaches go outside the respective models.

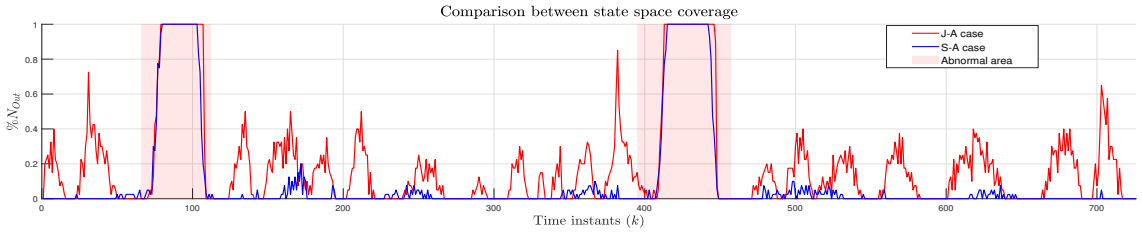
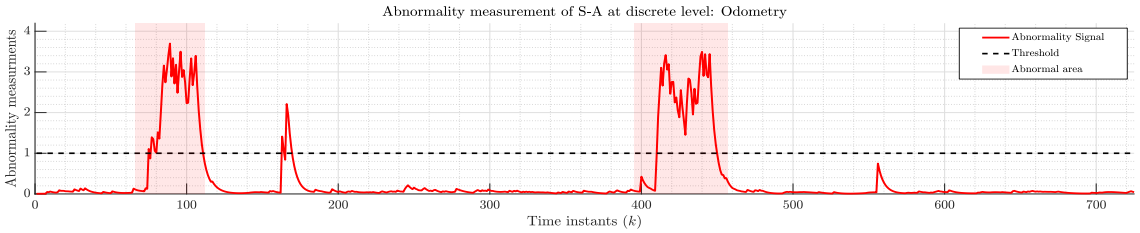


FIGURE 4.11: Percentage of particles that go out of the model's radius of acceptance at each time instant for J-A (red) and S-A (blue) protocols. Ground truth abnormality regions are indicated as red background rectangles.

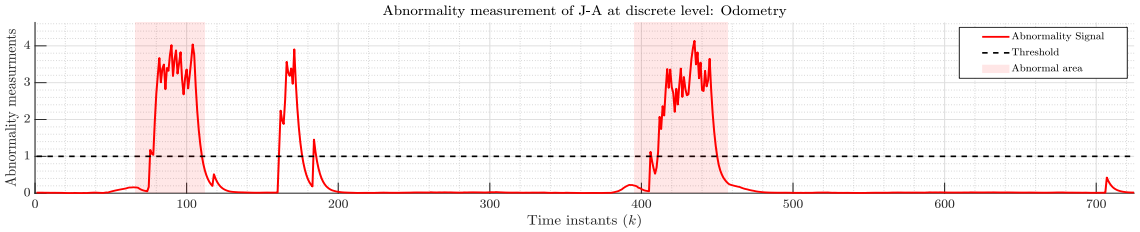
- ***Over-clustering results***. Similar to the under-clustering case, some results are displayed for the over-clustering compression level to show and compare the performances of S-A and J-A protocols when using a high number of words. Accordingly, Figure 4.12a and Figure 4.12b show the discrete abnormality signals from the odometry modality in S-A and J-A respectively. Note that AUC and ACC measurements associated with the odometry abnormality signals present similar performances in the over-clustering compression level, see last column in Table 4.1. Such a similar performance between S-A and J-a protocols is due to the high amount of words which often carry redundant information in the over-clustering case. When the number of words increases, models tend to be more precise and the coverage is larger. Nonetheless, a high number of discrete variables magnifies the complexity of the testing phase since large transition matrices encoding repetitive information are used for inference purposes.

  From Figure 4.12, it is possible to see how the odometry abnormality signals from S-A and J-A detect correctly both avoidance maneuvers (around $k = 100$ and $k = 430$) as anomalies. There is a false positive area around $k = 180$ which is identified by both protocols as an anomaly. Such a peak of abnormality is related to an abrupt curving behavior that does not match precisely with the curves observed previously in the training set. Despite that, the abnormality detection performance of both clustering protocols is quite satisfactory. Continuous abnormality signals related to the control

modality are provided in Figure 4.13. Two abnormal peaks are observed due to their generation when the avoidance maneuver takes place. These abnormalities refer to the regions where the vehicle do not go in a straight path while maneuvering. Specifically, Figure 4.13a presents some false abnormality detection parts (not as evident as the actual avoidance maneuver) associated with the curves executed by the vehicle. In the case shown in Figure 4.13b, regular curves are not detected as abnormalities, but some moments previous and after the avoidance are identified as abnormal.

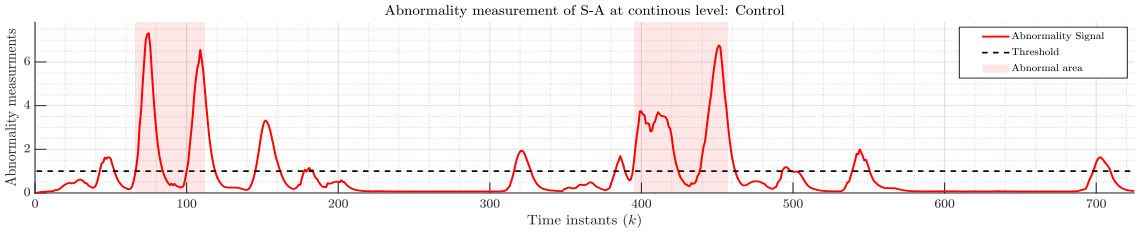


(A) Abnormality signal based on the odometry information in the S-A.

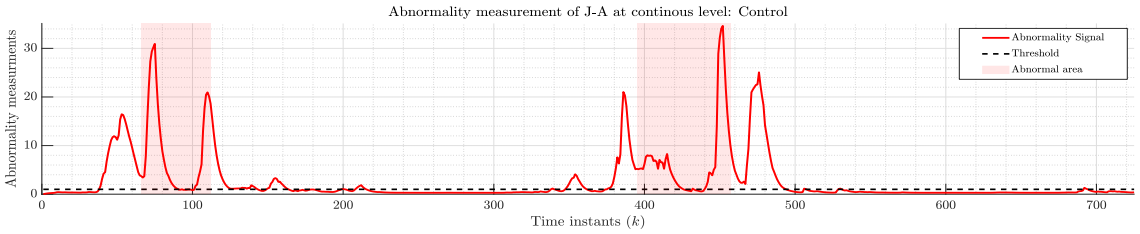


(B) Abnormality signal based on the odometry information in the J-A.

FIGURE 4.12: Discrete level abnormality signals (see Equation 4.22) of S-A and J-A protocols at the over-clustering compression level for odometry information.



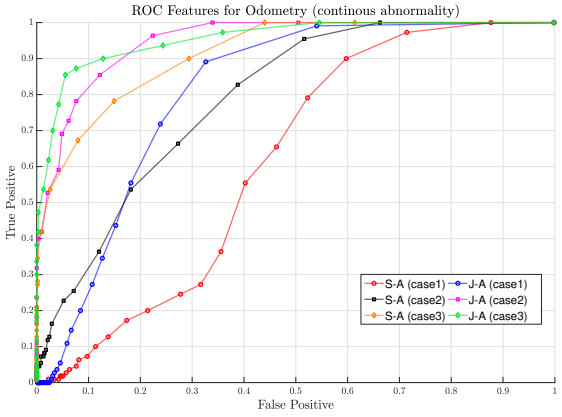
(A) Abnormality signal based on the control information in the S-A.



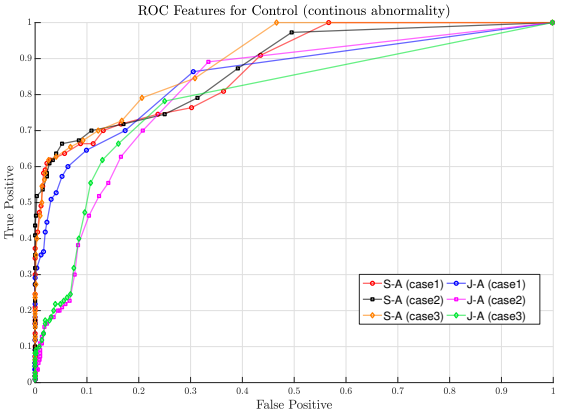
(B) Abnormality signal based on the control information in the J-A.

FIGURE 4.13: Continuous level abnormality signals (see Equation 4.21) of S-A and J-A protocols at the over-clustering compression level for control data.

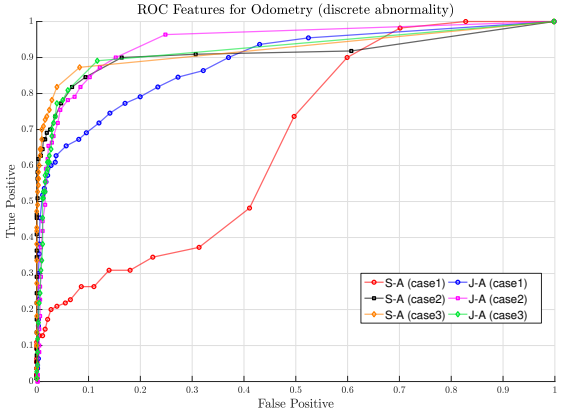
- ***ROC curves***. The rest of the section discusses the ROC curves from which AUC and ACC measurements in Table 4.3 are calculated. Accordingly, Figure 4.14 compares S-A and J-A protocols at detecting abnormalities based on continuous/discrete state information, i.e., by employing Equation 4.21 and Equation 4.22 respectively on odometry and control modalities.



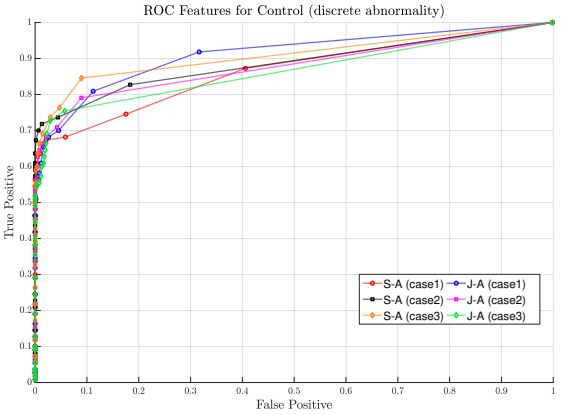
(A) ROC based on continuous abnormality measurements on odometry modality.



(B) ROC based on continuous abnormality measurements on control modality.



(C) ROC based on discrete abnormality measurements on odometry modality.



(D) ROC based on discrete abnormality measurements on control modality.

FIGURE 4.14: ROC curves that compare the S-A and J-A performances at detecting abnormalities in multimodal data at different clustering compression levels.

By analyzing curves in Figure 4.14a, it is possible to see how the performance at detecting anomalies in odometry data varies significantly when employing the S-A or J-A protocol. The J-A experiments based on continuous abnormalities clearly outperform the S-A ones. More specifically, it can be seen how under (case 1)/mid (case 2)-clustering compression levels in S-A produce a poor performance when compared with the respective levels in J-A. This result can be explained by the characteristics of the clusters in S-A,

which as explained previously, present a high coverage of GSs but do not offer precise dynamic models for prediction purposes.

When analyzing Figure 4.14c, it is possible to observe how the under-clustering compression level in S-A presents is a poor performance at detecting anomalies, whereas the same level in J-A performs quite well. It can be also seen how mid (case 2) and over clustering (case 3) compression levels in S-A and J-A perform with similar high accuracy. In such cases, since discrete components of clusters are employed for calculating the abnormality measurements (see equation 4.22) the S-A is able to compensate its disadvantages when predicting with its capabilities in covering GS-spaces.

Next, by comparing the different ROC curves in Figure 4.14b and Figure 4.14d, we can conclude that the control modality information is predicted quite well by the proposed abnormality measurements at all levels of compression from both clustering protocols. In other words, by considering a multi-modal approach that fuses odometry and control information, it is possible to obtain an accurate detection of abnormalities in the control data even with a low number of dictionary elements.

4.7.3 Discussions

Through this chapter, we proposed two different clustering protocols based on the processing and understanding of multi-modal data. Such protocols process information and build SA models based on an independent modality learning approach (i.e., S-A) and a direct combination of all multi-modal data at once (i.e., J-A). Obtained models encode GS information that is evaluated regarding the capability of explaining large amount of data (models' coverage) and estimating next future instants accurately (model's predictions and detection of abnormalities).

Our method has proven its ability to handle multi-modal information generated from a dynamical agent. A MPJF is employed by the two proposed approaches for making inferences related to future agent's GSs. Our proposed approaches allow the agent to combine multisensorial information for making estimations and detecting abnormalities at each time instant.

Data from a moving vehicle that executes different tasks in a controlled environment is employed for testing and validating the proposed method. Results suggest

that the J-A generates models that present a high prediction capabilities driving the J-A to be an attractive clustering protocol when making inferences on new multi-modal data that is similar to the training set. On the other hand, the S-A produces models with a high capability of explaining previously non-observed information and generating models that coverage extensive areas of the GS space. Such features make the S-A an attractive choice when dealing with new multi-modal data that do not follow the training data precisely. Based on the outcomes

Self-awareness model properties	**Description**
Generative modeling	Same as in Table 3.2.
Discriminative modeling	Same as in Table 3.2.
Hierarchical modeling	Same as in Table 3.2.
Temporal reasoning	Same as in Table 3.2.
Uncertain reasoning	Same as in Table 3.2.
Interactive	**Synchronization of proprioceptive and exteroceptive sensory information is employed for creating models that consider the agent's own internal and external states for embedding the interaction with its surroundings into the agent's knowledge. Interactive modeling enables decision-making exploiting contextual information.**

TABLE 4.4: Extended Properties of the proposed Self-awareness model.

of this chapter, the characteristics of the SA model are extended compared to those presented in Chapter 3 (see Table 3.2). Table 4.4 shows the common characteristics with those in Chapter 3 and the extra one highlighted.

Interactions between moving agents. The proposed work in this chapter is not only can be used for multi-modal data, but also it can be applied to track and interpret the interactions of multiple moving agents. Modeling interactions is based on the analysis of location data from different moving agents that modify their dynamics according to the rules of interactions. In particular, we use a J-A

based on two weights as pointed out in [99]. In this part, we validate the proposed method through a simulated data set introduced by [99].

The set of simulated data includes a moving agent, here called *follower*, chases another agent, here named *attractor*. In the training data, the motion of the follower is described by the velocity field shown in equation 4.28.

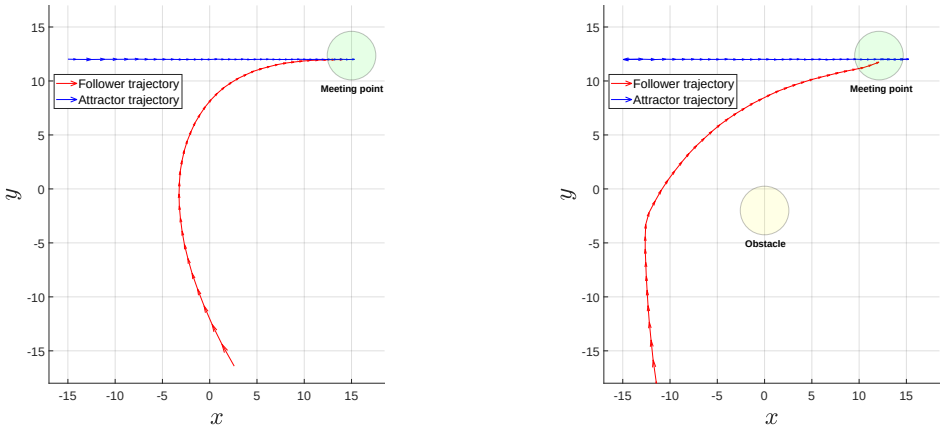
$$\vec{v}_f = \big(\psi + \frac{r^2}{\phi}\big)\hat{r} + \omega, \tag{4.28}$$

r represents the distance between both agents, ψ encodes the final speed with which the follower reaches the attractor, ϕ models the changes of follower's speed while it approaches the attractor, $\hat{r}$ is a unit vector that points at the attractor's location and $\omega \sim \mathcal{N}(0, \zeta)$.

The attractor motions consist of a horizontal dynamics along the x axis at a fixed height point y_{att}. Thus, the attractor can move in two senses: right or left inside the interval $[x_{att}^{(min)}, x_{att}^{(max)}]$. The attractor's dynamics is a continuous motion in one sense until it reaches an interval boundary when it starts moving in the opposite sense covering only the defined interval points. The speed of the attractor movements is defined as $|\vec{v}_a| = \Psi|\vec{v}_f|$, where $\Psi \in [0, 1)$ which guarantees that the follower reaches the attractor.

We use attractor-follower data that follow the rules described previously for learning a C-DBN structure. For simulation purposes, the following parameters are employed: $\psi = 0.85$, $\phi = 700$, $\zeta = 0.1$, $\Psi = 0.75$, $y_{att} = 12$, $x_{att}^{(min)} = -15$ and $x_{att}^{(max)} = 15$.

Abnormality detection. Testing trajectories are employed to detect abnormalities. Such new trajectories could follow exactly the same rules with which the C-DBN has been trained, or they could contain some changes induced to the presence of a static repulsive located in the center of the scene. Figure 4.15a and Figure 4.15b show normal (data that follows training set rules) and abnormal scenarios respectively. In both plots, red and blue arrows represent the trajectories of the follower and attractor agents. The final position of the attractor, i.e., when the follower reaches it, is displayed as a green circle. The repulsive agent is plotted as a yellow circle in Figure 4.15b.



(A) Normal interaction data. (B) Abnormal interaction data.

FIGURE 4.15: Trajectories of interacting agents.

Figures 4.16 and 4.17 show the result of abnormality detection in case of normal and abnormal interactions, Figures 4.15a and 4.15b correspondingly. As shown in Figure 4.16, we have a low abnormality (less than 0.1) which suggests that learned C-DBN understands the interacting rules of the simulator. From Figures 4.17, it is possible to see how high abnormality values are present in the initial portion of the trajectory data; such behavior (yellow background) is due to the repulsive agent's effects which alters the learned interaction model. Once the follower overpasses the obstacle, measurements of abnormality goes down (blue background) indicating that the follower-attractor interact according to the previously learned rules.

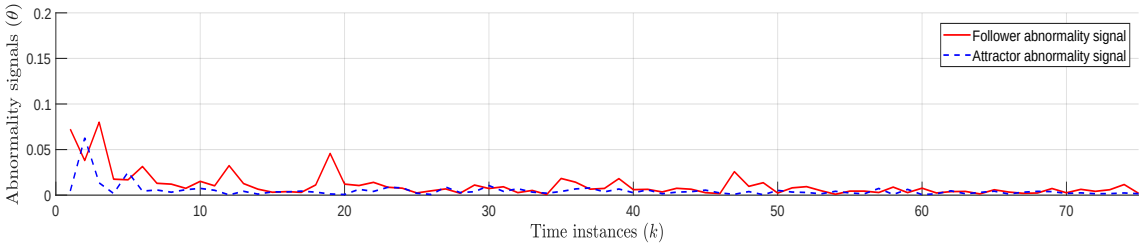


FIGURE 4.16: Results for normal agents' interaction.

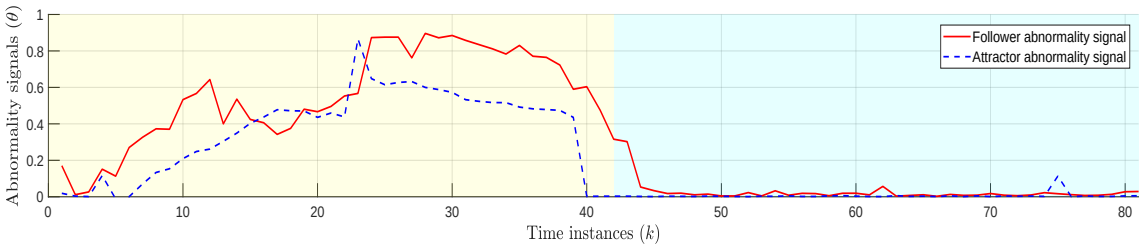
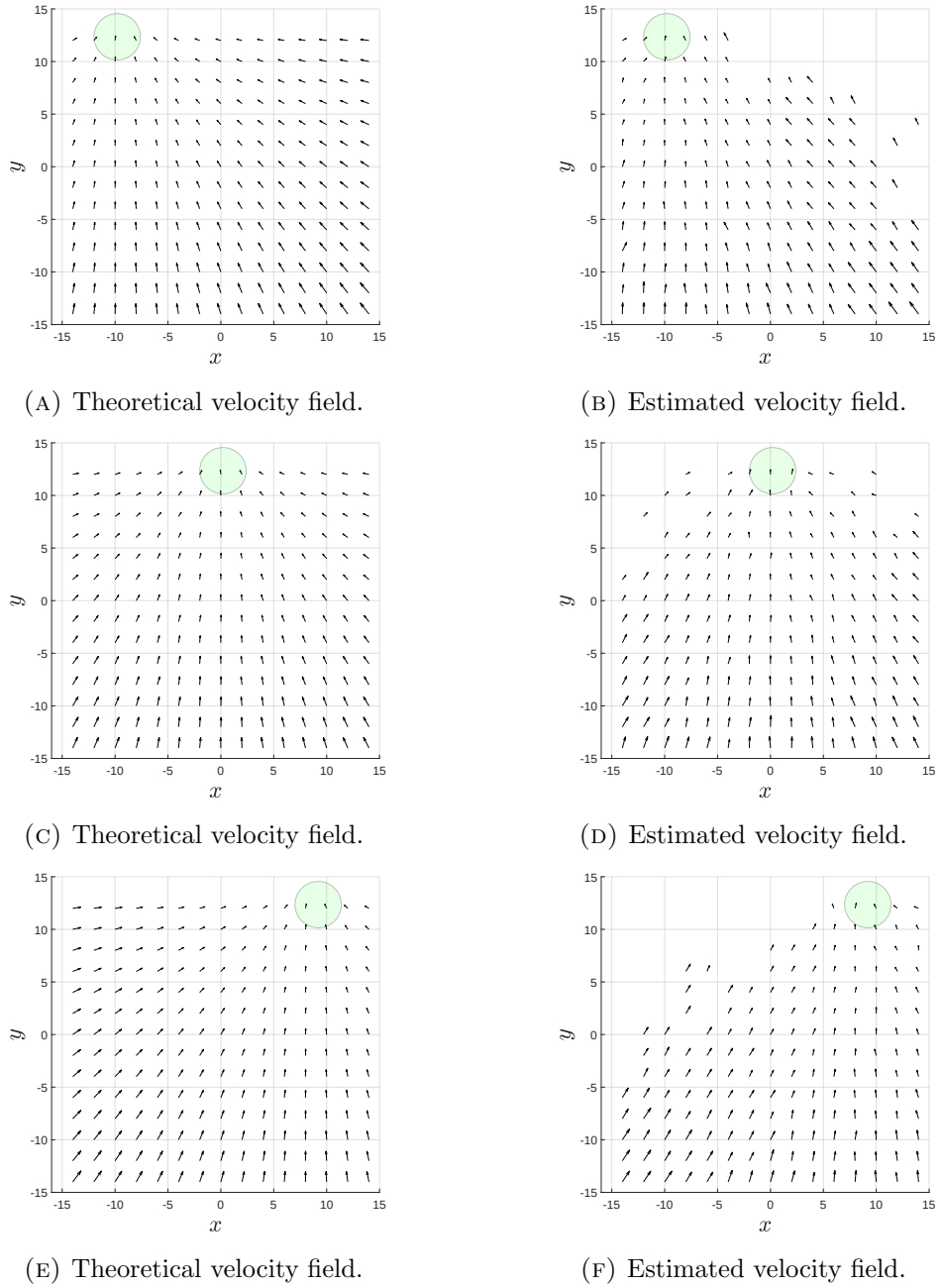


FIGURE 4.17: Results for abnormal agents' interaction.

Evaluation C-DBN. As the ground truth of the simulated rules is available, the latter provides a visual comparison between theoretical velocity fields and C-DBN motion estimations for different attractor-follower configurations.



(A) Theoretical velocity field.

(B) Estimated velocity field.

(C) Theoretical velocity field.

(D) Estimated velocity field.

(E) Theoretical velocity field.

(F) Estimated velocity field.

FIGURE 4.18: Theoretical and estimated velocity fields.

Figure 4.18 provides a qualitative comparison between theoretical velocity fields generated based on equation 4.28 and the corresponding prediction of the proposed DBN. Green circles represent the position of the attractor and arrows represent its generated velocity field. Empty spaces in C-DBN estimated fields are points

where the proposed method is not able to make predictions due to lack of evidence data in such areas.

Results suggest that attractive and repulsive forces can be modeled inside a C-DBN structure that codifies the normal behavior of observed agents. In what mentioned concerning the proposed method, the latter demonstrated the capability to encode the interactions of agents and employs such information for detecting anomalies due to previously unseen forces. Qualitative comparisons between the simulated and encoded C-DBN interaction rules are provided, demonstrating that the proposed models are capable of encoding observed behaviors into probability distributions.

Chapter 5

Conclusions and Future Work

In this thesis, we presented methods to learn an awareness model for an AA. For achieving aware artificial agents, we included a sense of SA (perceiving of own states) and situational awareness (understanding of external surrounding states) for the agent under study. Explored strategies have demonstrated that how PGMs, such as DBNs, can learn awareness models from multi-dimensional proprioceptive and exteroceptive signals acquired by the AA.

First, we show how to model the dynamics of an agent from a single positional modality in Chapter 2. This model can be used to represent situational awareness but from a single perspective. Accordingly, to enrich the information related to effects produced by the environment, we divided the agents' motions into a set of zones in the environment.

Each zone represents the activity of the agent in terms of a motivation where it moves towards a specific goal. For modelling such zones, we use a Bayesian reasoning representation for interpreting and modelling observed data. Such representation is used later on for further purposes such as classification, prediction and detection of abnormalities.

In order to increase the awareness of agents, we extended the work in Chapter 2 in Chapter 3. Since the single modality is inaccurate most of the time and an agent is considered full aware if it can dynamically observe itself and its environment through different sensorial modalities (proprioceptive and exteroceptive sensors) and learn a contextual representation by processing the observed multi-sensorial data. That is why in Chapter 3 we present a multi-modal SA incremental switching

model to perceive situations through different perspectives by considering multi-sensorial modalities that can be integrated to build a structure of cross-modal SA for an agent. Through this chapter, we showed that such a model could perform better since it uses multi-modal complementary information from different sources. Besides, modeling the SA helps the agent to understand its abilities and limitation for the sake of taking the actions accordingly.

The fusion of information from different sensors enhances the understanding of the agent itself and its surroundings and provides the basis for planning, decision making, and control. Starting from this fact, in Chapter 4, we introduced a multi-modal interaction model to model the causality between different sensory information. In contrast with Chapter 3, here we learn joint models between the different modalities in terms of their interactions. A coupled Bayesian network is used for representing the interaction at different levels (continuous and discrete). The latter step helps the agent to detect the abnormal/unseen situation from different perspectives jointly (as a uniform system).

We represented and modelled interactions among multi-sensory data as the last strategy in this work. By taking advantages of such approach with respect to other possible strategies, the former could be the potential for future paths that lead us to a complete SA model. Such model can be further developed to transfer the knowledge across different agents. Based on what is mentioned and the knowledge gained through the whole study, in the following we highlight the future work from our own perspective in a detailed manner.

In order to complete a full aware system, three main tasks are proposed for future work:

- ***Multi-DBNs****:* In this thesis, a coupled DBNs are considered as an initial model for learning the interaction between bi-modal data from an agent. However, the interaction models in the real-word scenarios are more complex. Hence, we consider more complex model to learn the interaction across different modalities. We propose a multi-DBN interaction model illustrated in Figure 5.1. The latter can model the interaction among three different modalities m, m' and m''. This model can be seen as a more general representation and more accurate way of our PGM model introduced in Chapter 4.

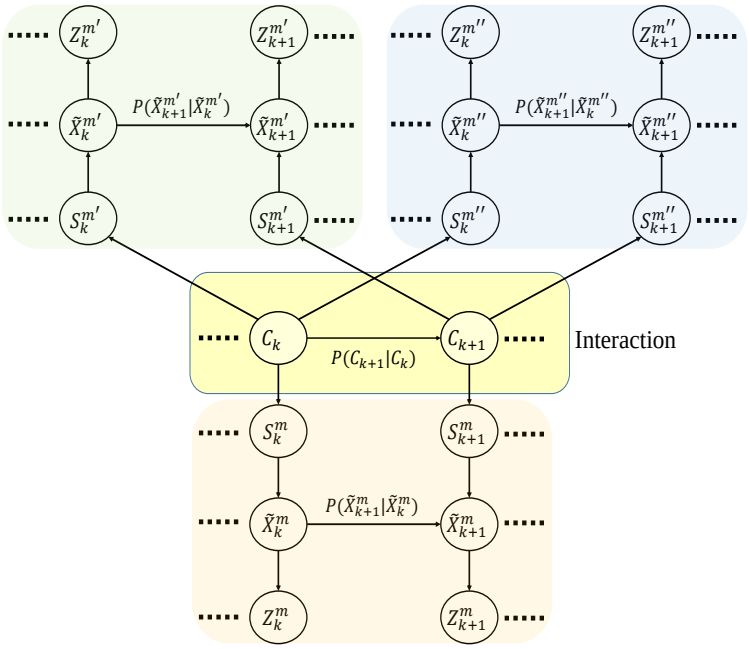


FIGURE 5.1: Proposed representation of Multi-DBNs for the SA.

- ***Knowledge transfer from agent to agent****:* Additionally, as future work it is also proposed to transfer the knowledge embedded in an agents' SA model into a totally different body. Transfer learning problem [100] has been studied in many machine learning-related tasks. In general, transfer learning is about storing knowledge while solving a problem and then using this knowledge for applying it to a different task [101], dataset [102] or another agent with a new body [103].

  In our scenario, the same set of rules should apply to other types of agents that intend to replicate observed behaviors of a given agent that accomplishes a given task. Then, when interaction models are learned, it is proposed to perform the respective mapping at the level of sensors, actuators, and interfaces in order to make a complete awareness transference possible.

  By obtaining a general representation of interactions, the latter is generic enough so that it can be used to transfer the learned knowledge into other agents. However, such knowledge transfer typically must be provided either by a full model of the tasks or by an explicit relation mapping one task into the other. From our point of view, We follow the second option where the mapping would represent a set of interactions.

  This idea can be used to create a methodology in which learning activities by one machine can be transferred to another one that looks at the first

one. From that viewpoint, the present work can be potentially used to build a cooperative/coupled framework among machines where they are able to transfer knowledge between them through observation and imitation of accomplished tasks.

- ***Decision making mechanism integration**:* The final goal of any AAs is to perform autonomously or act with a degree of autonomy. By using the probabilistic interaction model, the agent would be able to make decisions based on the observed situations and its internal states. In Figure 1.3, we proposed a fully AA where the decision block is a part of the system that takes as input the outputs of our SA models (i.e., predictions and abnormality measurements) to act accordingly.

  As we reviewed in this thesis, our SA model can provide information for the decision-making system. However, we did not use such information in a further step. A possible future path could be focusing on the development of such autonomous decision-making mechanism based on the SA model.

Bibliography

[1] P. Maes, "Designing autonomous agents: Theory and practice from biology to engineering and back," *Robotics and Autonomous Systems*, vol. 6, pp. 1–2, 1991. URL: https://books.google.it/books?id=cK-1pavJW98C

[2] P. Maes, "Artificial life meets entertainment: Lifelike autonomous agents," *Commun. ACM*, vol. 38, pp. 108–114, 1995. URL: https://doi.org/10.1145/219717.219808

[3] M. Huhn, J. P. Müller, J. Görmer, G. Homoceanu, N. Le, L. Märtin, C. Mumme, C. Schulz, N. Pinkwart, and C. Müller-Schloer, "Autonomous agents in organized localities regulated by institutions," in *IEEE International Conference on Digital Ecosystems and Technologies*, 2011, pp. 54–61. URL: https://doi.org/10.1109/DEST.2011.5936598

[4] B. Hayes-Roth, "An architecture for adaptive intelligent systems," *Artificial Intelligence*, vol. 72, p. 329–365, 1995. URL: https://doi.org/10.1016/0004-3702(94)00004-K

[5] A. Garro, M. Mühlhäuser, A. Tundis, S. Mariani, A. Omicini, and G. Vizzari, "Intelligent agents and environment," in *Reference Module in Life Sciences*. Elsevier, 2018. URL: http://www.sciencedirect.com/science/article/pii/B9780128096338203270

[6] N. R. Jennings and M. Wooldridge, "Applications of intelligent agents," in *Agent technology*. Springer, 1998, pp. 3–28. URL: https://doi.org/10.1007/978-3-662-03678-5_1

[7] N. J. Nilsson, "Human-level artificial intelligence? be serious!" *AI magazine*, vol. 26, pp. 68–75, 2005. URL: https://doi.org/10.1609/aimag.v26i4.1850

[8] J. Shabbir and T. Anwer, "Artificial intelligence and its role in near future," *ArXiv*, vol. abs/1804.01396, 2018. URL: https://arxiv.org/pdf/1804.01396

[9] A. Morin, "Levels of consciousness and self-awareness: A comparison and integration of various neurocognitive views," *Consciousness and Cognition*, vol. 15, pp. 358 – 371, 2006. URL: https://doi.org/10.1016/j.concog.2005.09.006

[10] A. Fenigstein, M. F. Scheier, and A. H. Buss, "Public and private self-consciousness: Assessment and theory," *Journal of Consulting and Clinical Psychology*, vol. 43, p. 522–527, 1975. URL: https://doi.org/10.1037/h0076760

[11] S. Baker, "The identification of the self." *Psychological Review*, vol. 4, pp. 272–284, 1897. URL: https://doi.org/10.1037/h0075515

[12] J. B. Asendorpf, V. Warkentin, and P.-M. Baudonnière, "Self-awareness and other-awareness. II: Mirror self-recognition, social contingency awareness, and synchronic imitation." *Developmental Psychology*, vol. 32, p. 313–321, 1996. URL: https://doi.org/10.1037/0012-1649.32.2.313

[13] J. Bajgar, J. Ciarrochi, R. Lane, and F. P. Deane, "Development of the levels of emotional awareness scale for children (leas-c)," *British Journal of Developmental Psychology*, vol. 23, pp. 569–586, 2005. URL: https://doi.org/10.1348/026151005X35417

[14] P. Rochat, "Five levels of self-awareness as they unfold early in life," *Consciousness and Cognition*, vol. 12, pp. 717 – 731, 2003. URL: https://doi.org/10.1016/S1053-8100(03)00081-3

[15] D. Hope and R. Heimberg, "Public and private self-consciousness and social phobia," *Journal of Personality Assessment*, vol. 52, pp. 626–639, 1988. URL: https://doi.org/10.1207/s15327752jpa5204_3

[16] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *IEEE Computer*, vol. 36, pp. 41–50, 2003. URL: https://doi.org/10.1109/MC.2003.1160055

[17] P. R. Lewis, M. Platzner, B. Rinner, J. Tørresen, and X. Yao, *Self-aware Computing Systems: An Engineering Approach*. Springer Publishing Company, Incorporated, 2016. URL: https://dl.acm.org/doi/book/10.5555/3001612

[18] A. Winfield, "Robots with internal models: a route to self-aware and hence safer robots," in *The Computer After Me*. Imperial College Press / World Scientific Book, 2014, pp. 237–252. URL: https://doi.org/10.1142/9781783264186_0016

[19] B. Rinner, L. Esterle, J. Simonjan, G. Nebehay, R. Pflugfelder, G. Fernandez, and P. R. Lewis, "Self-Aware and Self-Expressive Camera Networks," *IEEE Computer*, vol. 48, pp. 33–40, 2015. URL: https://doi.org/10.1109/MC.2015.209

[20] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, pp. 41–50, 2003. URL: https://doi.org/10.1109/MC.2003.1160055

[21] M. Möstl, J. Schlatow, R. Ernst, H. Hoffmann, A. Merchant, and A. Shraer, "Self-aware systems for the internet-of-things," in *International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, 2016, pp. 1–9. URL: https://ieeexplore.ieee.org/document/8462005/

[22] D. Kanapram, D. Campo, M. Baydoun, L. Marcenaro, E. L. Bodanese, C. Regazzoni, and M. Marchese, "Dynamic bayesian approach for decision-making in ego-things," in *IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019, pp. 909–914. URL: https://doi.org/10.1109/WF-IoT.2019.8767204

[23] M. Farrukh, A. Krayani, M. Baydoun, L. Marcenaro, Y. Gao, and C. S. Regazzoni, "Learning a switching bayesian model for jammer detection in the cognitive-radio-based internet of things," in *IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019, pp. 380–385. URL: https://doi.org/10.1109/WF-IoT.2019.8767187

[24] J. Schlatow, M. Moostl, R. Ernst, M. Nolte, I. Jatzkowski, M. Maurer, C. Herber, and A. Herkersdorf, "Self-awareness in autonomous automotive systems," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2017, pp. 1050–1055. URL: https://doi.org/10.23919/DATE.2017.7927145

[25] S. Spaccapietra, C. Parent, M. L. Damiani, J. A. de Macedo, F. Porto, and C. Vangenot, "A conceptual view on trajectories," *Data*

and Knowledge Engineering, vol. 65, pp. 126 – 146, 2008. URL: http://www.sciencedirect.com/science/article/pii/S0169023X07002078

[26] Y. Hu, K. Janowicz, D. Carral, S. Scheider, W. Kuhn, G. Berg-Cross, P. Hitzler, M. Dean, and D. Kolas, "A geo-ontology design pattern for semantic trajectories," *International conference on spatial information theory*, vol. 8116 LNCS, pp. 438–456, 2013. URL: https://doi.org/10.1007/978-3-319-01790-7_24

[27] C. Parent, S. Spaccapietra, C. Renso, G. Andrienko, N. Andrienko, V. Bogorny, M. L. Damiani, A. Gkoulalas-Divanis, J. Macedo, N. Pelekis, Y. Theodoridis, and Z. Yan, "Semantic trajectories modeling and analysis," *ACM Computing Surveys*, vol. 45, pp. 42:1–42:32, 2013. URL: http://doi.acm.org/10.1145/2501654.2501656

[28] R. Das and S. Winter, "Automated urban travel interpretation: A bottom-up approach for trajectory segmentation," *Sensors*, vol. 16, p. 1962, 2016. URL: https://doi.org/10.3390/s16111962

[29] D. Campo, A. Betancourt, L. Marcenaro, and C. Regazzoni, "Static force field representation of environments based on agents' nonlinear motions," *Eurasip Journal on Advances in Signal Processing*, vol. 2017, p. 13, 2017. URL: https://doi.org/10.1186/s13634-017-0444-5

[30] F. Castaldo, F. A. N. Palmieri, and C. S. Regazzoni, "Bayesian analysis of behaviors and interactions for situation awareness in transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, pp. 313–322, 2016. URL: https://doi.org/10.1109/TITS.2015.2466695

[31] L. Snidaro, J. García, J. Llinas, and E. Blasch, *Context-Enhanced Information Fusion: Boosting Real-World Performance with Domain Knowledge*. Springer International Publishing, 2016. URL: https://www.bookdepository.com/Context-Enhanced-Information-Fusion-Jesus-Garcia/9783319289694

[32] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, pp. 1018–1021, 2010. URL: https://doi.org/10.1126/science.1177170

[33] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi, "Human mobility, social ties, and link prediction," in *Proceedings of the*

17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ser. KDD '11. ACM, 2011, pp. 1100–1108. URL: http://doi.acm.org/10.1145/2020408.2020581

[34] G. Pallotta, M. Vespe, and K. Bryan, "Vessel pattern knowledge discovery from ais data: A framework for anomaly detection and route prediction," *Entropy*, vol. 15, pp. 2218–2245, 2013. URL: https://doi.org/10.3390/e15062218

[35] J. V. Benavides, J. Kaneshige, S. Sharma, R. Panda, and M. Steglinski, "Implementation of a trajectory prediction function for trajectory based operations," in *AIAA Atmospheric Flight Mechanics Conference*, 2014, p. 2198. URL: https://doi.org/10.2514/6.2014-2198

[36] R. Bar-David and M. Last, "Context-aware location prediction," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9546, pp. 165–185, 2016. URL: https://doi.org/10.1007/978-3-319-29009-6_9

[37] M. Veres and M. Moussa, "Deep learning for intelligent transportation systems: A survey of emerging trends," *IEEE Transactions on Intelligent Transportation Systems (TITS)*, pp. 1–17, 2019. URL: https://doi.org/10.1109/TITS.2019.2929020

[38] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE Transactions on Intelligent Vehicle*, p. 26, 2019. URL: http://arxiv.org/abs/1906.05113

[39] M. M. Murray, A. Thelen, G. Thut, V. Romei, R. Martuzzi, and P. J. Matusz, "The multisensory function of the human primary visual cortex," *Neuropsychologia*, vol. 83, pp. 161 – 169, 2016. URL: https://doi.org/10.1016/j.neuropsychologia.2015.08.011

[40] G. Soter, A. Conn, H. Hauser, and J. Rossiter, "Bodily aware soft robots: Integration of proprioceptive and exteroceptive sensors," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2448–2453. URL: https://doi.org/10.1109/ICRA.2018.8463169

[41] M. M. Murray, D. J. Lewkowicz, A. Amedi, and M. T. Wallace, "Multisensory processes: A balancing act across the lifespan," *Trends in*

Neurosciences, vol. 39, pp. 567 – 579, 2016. URL: https://doi.org/10.1016/j.tins.2016.05.003

[42] P. C. Stacey, P. T. Kitterick, S. D. Morris, and C. J. Sumner, "The contribution of visual information to the perception of speech in noise with and without informative temporal fine structure," *Hearing Research*, vol. 336, pp. 17 – 28, 2016. URL: https://doi.org/10.1016/j.heares.2016.04.002

[43] C. L. Blackburn, P. T. Kitterick, G. Jones, C. J. Sumner, and P. C. Stacey, "Visual speech benefit in clear and degraded speech depends on the auditory intelligibility of the talker and the number of background talkers," *Trends in Hearing*, vol. 23, pp. 1 – 14, 2019. URL: https://doi.org/10.1177/2331216519837866

[44] C. Spence, "Multisensory flavor perception," *Cell*, vol. 161, pp. 24 – 35, 2015. URL: http://www.sciencedirect.com/science/article/pii/S0092867415002603

[45] J. Prescott, "Multisensory processes in flavour perception and their influence on food choice," *Current Opinion in Food Science*, vol. 3, pp. 47 – 52, 2015. URL: https://doi.org/10.1016/j.cofs.2015.02.007

[46] B. Stein, *The New Handbook of Multisensory Processing*, ser. The MIT Press. MIT Press, 2012. URL: https://books.google.it/books?id=tfo9jwEACAAJ

[47] A. Hammond-Kenny, V. M. Bajo, A. J. King, and F. R. Nodal, "Behavioural benefits of multisensory processing in ferrets," *European Journal of Neuroscience*, vol. 45, pp. 278–289, 2017. URL: https://doi.org/10.1111/ejn.13440

[48] L. Freeman, K. C. Wood, and J. K. Bizley, "Multisensory stimuli improve relative localisation judgments compared to unisensory auditory or visual stimuli," *The Journal of the Acoustical Society of America*, vol. 143, pp. 516 – 522, 2018. URL: https://doi.org/10.1101/268540

[49] P. J. Matusz, M. T. Wallace, and M. M. Murray, "A multisensory perspective on object memory," *Neuropsychologia*, vol. 105, pp. 243 – 252, 2017. URL: https://doi.org/10.1016/j.neuropsychologia.2017.04.008

[50] L. Shams and A. R. Seitz, "Benefits of multisensory learning," *Trends in Cognitive Sciences*, vol. 12, pp. 411 – 417, 2008. URL: https://doi.org/10.1016/j.tics.2008.07.006

[51] A. Thelen, P. J. Matusz, and M. M. Murray, "Multisensory context portends object memory," *Current Biology*, vol. 24, pp. R734 – R735, 2014. URL: https://doi.org/10.1016/j.cub.2014.06.040

[52] C. Fetsch, A. Pouget, G. Deangelis, and D. E Angelaki, "Neural correlates of reliability-based cue weighting during multisensory integration," *Nature neuroscience*, vol. 15, pp. 146–54, 2011. URL: https://doi.org/10.1038/nn.2983

[53] M. Dumitru, A. Pasqualotto, and A. Myachykov, *Multisensory Integration: Brain, Body and the World*. Frontiers Media SA, 2016, vol. 6. URL: https://doi.org/10.3389/fpsyg.2015.02046

[54] R. Chatila, E. Renaudo, M. Andries, R.-O. Chavez-Garcia, P. Luce-Vayrac, R. Gottstein, R. Alami, A. Clodic, S. Devin, B. Girard, and M. Khamassi, "Toward self-aware robots," *Frontiers Robotics AI*, vol. 5, p. 88, 2018. URL: https://doi.org/10.3389/frobt.2018.00088

[55] A. E. Martin, "Language processing as cue integration: Grounding the psychology of language in perception and neurophysiology," *Frontiers in psychology*, vol. 7, p. 120, 2016. URL: https://doi.org/10.3389/fpsyg.2016.00120

[56] P. Daras, S. Manolopoulou, and A. Axenopoulos, "Search and retrieval of rich media objects supporting multiple multimodal queries," *IEEE Transactions on Multimedia*, vol. 14, pp. 734–746, 2012. URL: https://doi.org/10.1109/TMM.2011.2181343

[57] X. Hu, K. Li, J. Han, X. Hua, L. Guo, and T. Liu, "Bridging the semantic gap via functional brain imaging," *IEEE Transactions on Multimedia*, vol. 14, pp. 314–325, 2012. URL: https://doi.org/10.1109/TMM.2011.2172201

[58] J. S. Brown, A. Collins, and G. Harris, "Artificial intelligence and learning strategies," in *Learning strategies*. Elsevier, 1978, pp. 107–139. URL: https://doi.org/10.1016/B978-0-12-526650-5.50010-1

[59] O. Ostapenko, M. Puscas, T. Klein, P. Jahnichen, and M. Nabi, "Learning to remember: A synaptic plasticity driven framework for continual learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 321–11 329. URL: http://arxiv.org/abs/1904.03137

[60] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, 2019. URL: https://doi.org/10.1016/j.neunet.2019.01.012

[61] D. Rolnick, A. Ahuja, J. Schwarz, T. P. Lillicrap, and G. Wayne, "Experience replay for continual learning," *Advances in Neural Information Processing Systems 32*, pp. 348–358, 2018. URL: https://arxiv.org/abs/1811.11682v1

[62] J. Schmidhuber, "A general method for multi-agent reinforcement learning in unrestricted environments," in *Adaptation, Coevolution and Learning in Multiagent Systems*, 1996, pp. 84–87. URL: https://www.aaai.org/Papers/Symposia/Spring/1996/SS-96-01/SS96-01-016.pdf

[63] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 2990–2999. URL: https://arxiv.org/pdf/1705.08690.pdf

[64] S. Thrun and L. Pratt, *Learning to learn*. Springer Science & Business Media, 2012. URL: https://doi.org/10.1007/978-1-4615-5529-2

[65] D. Campo, M. Baydoun, L. Marcenaro, and C. S. Regazzoni, "Task-dependent saliency estimation from trajectories of agents in video sequences," in *IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 4252–4256. URL: https://doi.org/10.1109/ICIP.2017.8297084

[66] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003. URL: https://dl.acm.org/doi/book/10.5555/773294

[67] G. Piriou, P. Bouthemy, and J.-F. Yao, "Recognition of dynamic video contents with global probabilistic models of visual motion," *IEEE Transactions on Image Processing*, vol. 15, pp. 3417–3430, 2006. URL: https://doi.org/10.1109/TIP.2006.881963

[68] D. Campo, V. Bastani, L. Marcenaro, and C. Regazzoni, "Incremental learning of environment interactive structures from trajectories of individuals," in *19th International Conference on Information Fusion*, 2016, pp. 589–596. URL: https://ieeexplore.ieee.org/document/7527941

[69] D. Campo, M. Baydoun, P. Marin, D. Martin, L. Marcenaro, A. de la Escalera, and C. Regazzoni, "Learning probabilistic awareness models for detecting abnormalities in vehicle motions," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2019. URL: https://doi.org/10.1109/TITS.2019.2909980

[70] M. Baydoun, D. Campo, V. Sanguineti, L. Marcenaro, A. Cavallaro, and C. Regazzoni, "Learning switching models for abnormality detection for autonomous driving," in *21st International Conference on Information Fusion*, 2018, pp. 2606–2613. URL: https://doi.org/10.23919/ICIF.2018.8455592

[71] D. Campo, M. Baydoun, L. Marcenaro, A. Cavallaro, and C. S. Regazzoni, "Modeling and classification of trajectories based on a gaussian process decomposition into discrete components," in *14th IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2017, pp. 1–6. URL: https://doi.org/10.1109/AVSS.2017.8078495

[72] D. Campo, M. Baydoun, L. Marcenaro, A. Cavallaro, and C. S. Regazzoni, "Unsupervised trajectory modeling based on discrete descriptors for classifying moving objects in video sequences," in *25th IEEE International Conference on Image Processing*, 2018, pp. 833–837. URL: https://doi.org/10.1109/ICIP.2018.8451837

[73] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, pp. 4282–4286, 1995. URL: https://doi.org/10.1103/PhysRevE.51.4282

[74] Z. Li and J. Chen, "Superpixel segmentation using linear spectral clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1356–1363. URL: https://doi.org/10.1109/CVPR.2015.7298741

[75] T. Kailath, "The divergence and bhattacharyya distance measures in signal selection," *IEEE Transactions on Communication Technology*, vol. 15, pp. 52–60, 1967. URL: https://doi.org/10.1109/TCOM.1967.1089532

[76] P. Marın-Plaza, J. Beltrán, A. Hussein, B. Musleh, D. Martın, A. de la Escalera, and J. M. Armingol, "Stereo vision-based local occupancy grid map for autonomous navigation in ros," in *11th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, vol. 4, 2016. URL: https://doi.org/10.5220/0005787007010706

[77] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, 2014. URL: https://doi.org/10.15607/RSS.2014.X.007

[78] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, 2009, p. 5. URL: http://www.willowgarage.com/sites/default/files/icraoss09-ROS.pdf

[79] B. Morris and M. Trivedi, "Learning trajectory patterns by clustering: Experimental studies and comparative evaluation," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 312–319. URL: https://doi.org/10.1109/CVPR.2009.5206559

[80] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[81] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *8th European Conference on Computer Vision*, T. Pajdla and J. Matas, Eds., 2004, pp. 25–36. URL: https://rdcu.be/b01yY

[82] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5967–5976. URL: https://doi.org/10.1109/CVPR.2017.632

[83] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. S. Regazzoni, and N. Sebe, "Abnormal event detection in videos using generative adversarial nets," in *IEEE International Conference on Image Processing*, 2017, pp. 1577–1581. URL: https://doi.org/10.1109/ICIP.2017.8296547

[84] A. Mazzu, P. Morerio, L. Marcenaro, and C. S. Regazzoni, "A cognitive control-inspired approach to object tracking," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2697–2711, 2016. URL: https://doi.org/10.1109/TIP.2016.2553781

[85] C. E. Antoniak, "Mixtures of dirichlet processes with applications to bayesian nonparametric problems," *The annals of statistics*, vol. 2, pp. 1152–1174, 1974. URL: https://www.jstor.org/stable/2958336

[86] J. Sethuraman, "A constructive definition of dirichlet priors," *Statistica sinica*, vol. 4, pp. 639–650, 1994. URL: https://www.jstor.org/stable/24305538

[87] D. J. Aldous, "Exchangeability and related topics," in *École d'Été de Probabilités de Saint-Flour XIII—1983*. Springer, 1985, pp. 1–198. URL: https://link.springer.com/chapter/10.1007%2FBFb0099421

[88] T. Kohonen, *Self-Organizing Maps*, ser. Physics and astronomy online library. Springer Berlin Heidelberg, 2001, vol. 30. URL: https://doi.org/10.1007/978-3-642-56927-2

[89] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," *International Conference on Learning Representations*, 2017. URL: http://arxiv.org/abs/1701.04862

[90] V. Bastani, L. Marcenaro, and C. Regazzoni, "Online nonparametric bayesian activity mining and analysis from surveillance video," *IEEE Transactions on Image Processing*, vol. 25, pp. 2089–2102, 2016. URL: https://doi.org/10.1109/TIP.2016.2540813

[91] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks," in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, 2000, pp. 176–183. URL: https://doi.org/10.1007/978-1-4757-3437-9_24

[92] A. Doucet, N. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump markov linear systems," *IEEE Transactions on Signal Processing*, vol. 49, pp. 613–624, 2001. URL: https://doi.org/10.1109/78.905890

[93] K. J. Friston, B. Sengupta, and G. Auletta, "Cognitive dynamics: From attractors to active inference," *Proceedings of the IEEE*, vol. 102, pp. 427–445, 2014. URL: https://doi.org/10.1109/JPROC.2014.2306251

[94] S. Chung, J. Lim, K. J. Noh, G. Kim, and H. Jeong, "Sensor data acquisition and multimodal sensor fusion for human activity recognition using deep learning," *Sensors*, vol. 19, p. 1716, 2019. URL: https://doi.org/10.3390/s19071716

[95] S. Park, C. Meeker, L. Weber, L. Bishop, J. Stein, and M. Ciocarlie, "Multimodal sensing and interaction for a robotic hand orthosis," *IEEE Robotics and Automation Letters*, vol. 4, pp. 315–322, 2019. URL: https://doi.org/10.1109/LRA.2018.2890199

[96] S. Nedelkoski, J. Cardoso, and O. Kao, "Anomaly detection from system tracing data using multimodal deep learning," in *IEEE 12th International Conference on Cloud Computing (CLOUD)*, 2019, pp. 179–186. URL: https://doi.org/10.1109/CLOUD.2019.00038

[97] H. Iqbal, D. Campo, M. Baydoun, L. Marcenaro, D. M. Gomez, and C. Regazzoni, "Clustering optimization for abnormality detection in semi-autonomous systems," in *1st International Workshop on Multimodal Understanding and Learning for Embodied Applications*. ACM, 2019, pp. 33–41. URL: https://doi.org/10.1145/3347450.3357657

[98] Kihwan Kim, Dongryeol Lee, and I. Essa, "Gaussian process regression flow for analysis of motion trajectories," in *International Conference on Computer Vision*, 2011, pp. 1164–1171. URL: https://doi.org/10.1109/ICCV.2011.6126365

[99] M. Baydoun, D. Campo, D. Kanapram, L. Marcenaro, and C. S. Regazzoni, "Prediction of multi-target dynamics using discrete descriptors: an interactive approach," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3342–3346. URL: https://doi.org/10.1109/ICASSP.2019.8682272

[100] S. P. Singh, "Transfer of learning by composing solutions of elemental sequential tasks," *Machine Learning*, vol. 8, pp. 323–339, 1992. URL: https://doi.org/10.1007/BF00992700

[101] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4068–4076. URL: https://doi.org/10.1109/ICCV.2015.463

[102] D. Stamos, S. Martelli, M. Nabi, A. McDonald, V. Murino, and M. Pontil, "Learning with dataset bias in latent subcategory models," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3650–3658. URL: https://doi.org/10.1109/CVPR.2015.7298988

[103] G. Boutsioukis, I. Partalas, and I. Vlahavas, "Transfer learning in multi-agent reinforcement learning domains," in *Recent Advances in Reinforcement Learning*, S. Sanner and M. Hutter, Eds. Springer Berlin Heidelberg, 2012, pp. 249–260. URL: https://doi.org/10.1007/978-3-642-29946-9_25